Scientific Excellence • Resource Protection & Conservation • Benefits for Canadians
Excellence scientifique • Protection et conservation des ressources • Bénéfices aux Canadiens

System Documentation for the Mark Recovery Program

M. A. Holmes and J. M. Hamer

Department of Fisheries and Oceans
Fisheries Research Branch
Pacific Biological Station
Nanaimo, British Columbia V9R 5K6

June 1988

Canadian Technical Report of Fisheries and Aquatic Sciences No. 1601

Fisheries and Oceans Pêches et Océans

Canada

89-00784/

Canadian Technical Report of Fisheries and Aquatic Sciences

Technical reports contain scientific and technical information that contributes to existing knowledge but which is not normally appropriate for primary literature. Technical reports are directed primarily toward a worldwide audience and have an international distribution. No restriction is placed on subject matter and the series reflects the broad interests and policies of the Department of Fisheries and Oceans, namely, fisheries and aquatic sciences.

Technical reports may be cited as full publications. The correct citation appears above the abstract of each report. Each report is abstracted in *Aquatic Sciences and Fisheries Abstracts* and indexed in the Department's annual index to scientific and technical publications.

Numbers 1–456 in this series were issued as Technical Reports of the Fisheries Research Board of Canada. Numbers 457–714 were issued as Department of the Environment, Fisheries and Marine Service, Research and Development Directorate Technical Reports. Numbers 715–924 were issued as Department of Fisheries and the Environment, Fisheries and Marine Service, Technical Reports. The current series name was changed with report number 925.

Technical reports are produced regionally but are numbered nationally. Requests for individual reports will be filled by the issuing establishment listed on the front cover and title page. Out-of-stock reports will be supplied for a fee by commercial agents.

Rapport technique canadien des sciences halieutiques et aquatiques

Les rapports techniques contiennent des renseignements scientifiques et techniques qui constituent une contribution aux connaissances actuelles, mais qui ne sont pas normalement appropriés pour la publication dans un journal scientifique. Les rapports techniques sont destinés essentiellement à un public international et ils sont distribués à cet échelon. Il n'y a aucune restriction quant au sujet; de fait, la série reflète la vaste gamme des intérêts et des politiques du ministère des Pêches et des Océans, c'est-à-dire les sciences halieutiques et aquatiques.

Les rapports techniques peuvent être cités comme des publications complètes. Le titre exact paraît au-dessus du résumé de chaque rapport. Les rapports techniques sont résumés dans la revue *Résumés des sciences aquatiques et halieutiques*, et ils sont classés dans l'index annuel des publications scientifiques et techniques du Ministère.

Les numéros 1 à 456 de cette série ont été publiés à titre de rapports techniques de l'Office des recherches sur les pêcheries du Canada. Les numéros 457 à 714 sont parus à titre de rapports techniques de la Direction générale de la recherche et du développement, Service des pêches et de la mer, ministère de l'Environnement. Les numéros 715 à 924 ont été publiés à titre de rapports techniques du Service des pêches et de la mer, ministère des Pêches et de l'Environnement. Le nom actuel de la série a été établi lors de la parution du numéro 925.

Les rapports techniques sont produits à l'échelon régional, mais numérotés à l'échelon national. Les demandes de rapports seront satisfaites par l'établissement auteur dont le nom figure sur la couverture et la page du titre. Les rapports épuisés seront fournis contre rétribution par des agents commerciaux.

Canadian Technical Report of
Fisheries and Aquatic Sciences No. 1601

June 1988

SYSTEM DOCUMENTATION FOR THE MARK RECOVERY PROGRAM

by

M. A. Holmes and J. M. Hamer

Department of Fisheries and Oceans
Fisheries Research Branch
Pacific Biological Station
Nanaimo, British Columbia V9R 5K6

(c) Minister of Supply and Services Canada 1988

Cat. No. Fs 97-6/1601E ISSN 0706-6457

Correct citation for this publication:

Holmes, M. A. and J. M. Hamer. 1988. System documentation for the mark recovery program. Can. Tech. Rep. Fish. Aquat. Sci. 1601: 103 p.

TABLE OF CONTENTS

	Page
ABSTRACT	v
INTRODUCTION	1
SECTION I	1
BACKGROUND	1
SYSTEM DESIGN	1
DATA STRUCTURE	2
REPORTS	2
UPDATES	2
SECTION II	3
COMMON BLOCKS	3
Introduction	3
Cwt Release Common Block	4
Finclip Release Common Block	6
Commercial Recovery Common Block	9
SOURCE CODE MODULES	11
Introduction	11
Commercial Sample Source Code	11
Escapement Sample Source Code	12
Random Recovery Common Block	13
Sport Recovery Matrix	14
Sport Mark Incidence Counts	15
TABLES	16
Introduction	16
Table Type Chart	16
TABLE DESCRIPTIONS	22
Production Area Table	22
Catch Region Name Table	24
Sport Catch Region Table	26
SUBROUTINE OVERVIEW	28
Introduction	28
Subroutines	28
Subroutine Hierarchy	31
Subroutines and Restricted Numbers	33
CWTAJIM	34
CWTCALWEK	35
CWTCHAIN	37
CWTCMSTCR	39
CWTCOLOUR	41
CWTCRWK	42
CWTCSGET	44
CWTESGET	45
CWTGRNAM	47
CWTHISITE	48
CWTMI1	49

TABLE OF CONTENTS (cont'd)

	Page
CWTMI2	51
CWTMMW	53
CWTNONGET	55
CWTNONOPN	57
CWTNONTAG	58
CWTOPCHN	59
CWTOPMST	60
CWTPAREA	61
CWTPROGTYP	63
CWTREARTYP	65
CWTRECRD	66
CWTREGNAM	68
CWTRELGET	70
CWTRELRD	72
CWTSEXMAT	73
CWTSITGET	74
CWTSML	76
CWTSPSTCR	77
CWTSTAGE	79
CWTSTKTYP	81
CWTSUM1	82
FINOPMST	84
FINRELGET	85
FINRELRD	87
MICROTRAN	88
REGIONGR	90
RELSTAGE	91
RUNTYPE	92
SAMLOC	93
SPECIES	94
STATSUB	96
STUDYTYPE	98
ACKNOWLEDGEMENTS	99
APPENDIX I	100
APPENDIX II	101
APPENDIX III	102

ABSTRACT

Holmes, M. A. and J. M. Hamer. 1988. System documentation for the mark recovery program. Can. Tech. Rep. Fish. Aquat Sci. 1601: 103 p.

The Mark Recovery Program (MRP) database of the Pacific Region, Canadian Department of Fisheries and Oceans is maintained by the Biological Science Branch (BSB) in Nanaimo, B.C. (Pacific Biological Station). This database consists of hatchery production and coded-wire tag (Jefferts et al. 1963) releases from all fishery agencies (California through S.E. Alaska), recoveries of coded-wire tag (CWT) salmonids in all Pacific coastal fisheries since 1975, and sampling information related to recovery effort. The database also utilizes historical catch data from the BSB database on commercial salmon catches (Wong 1983) and salmon sport catch and sampling data provided by the Fisheries Branch. Information is accessible via menu driven report generators for commonly requested data, a spreadsheet program to select specific information, and/or via user written programs employing the MRP subroutine library. This report documents user accessible subroutines, common blocks, source code and example shell programs.

RÉSUMÉ

Holmes, M. A. and J. M. Hamer. 1988. System documentation for the mark recovery program. Can. Tech. Rep. Fish. Aquat Sci. 1601: 103 p.

La base de données du programme de récupération des étiquettes de la Région du Pacifique (ministère des Pêches et des Océans du Canada) est tenue à jour par la Direction des sciences biologiques à NanaÏmo (Station de biologie du Pacifique, C.-B.). Cette base de données comprend les données sur la production des pisciculture et les mises à l'eau de poissons marqués par fils codés (Jefferts et coll. 1963) effectuées par tous les organismes de gestion des pêches (Californie au sud-est de l'Alaska), les récupérations de salmonidés portant des fils codés de toutes les pêches côtières du Pacifique depuis 1975 et les prélèvements relatifs à l'effort de pêche. La base de données englobe aussi les données historiques sur les prises de saumons de la pêche commerciale (Wong 1983) et de la pêche sportive de la Direction des sciences biologiques, de même que les données sur les prélèvements fournies par la Direction des pêches. Cette information peut être obtenue par programmes d'édition à base de menus, pour les données couramment demandées, par programmes de calcul de tableaux, pour des renseignements plus spécifiques, ou par programmes rédigés par les utilisateurs basés sur la bibliothèque de sous-programmes. Le présent rapport traite des sous-programmes accessibles aux utilisateurs, des blocs communs et du code de sources, et donne des exemples de programmes cadres.

Mots clés: Programme de récupération des étiquettes, base de données, fils codés, salmonidés, échantillonnage, données historiques

INTRODUCTION

The system described in this report is resident on the VAX at PBS and is available to all users. Sampling and data entry of commercial and sport data are done under FB managed contracts, while escapement data entry and database management is handled by BSB. System architecture was designed by J. M. Hamer and L. A. Lapi while the research, development and implementation of the database was handled by a number of interested user groups. Currently system problems are handled by Marc Hamer. Data is handled by the Salmon Stock Assessment group.

The intent of this report is to provide programming documentation of subroutines, tables and common blocks needed for accessing the MRP database. Section I outlines past systems, present system design, data flow, pointers and reports. Section II documents common blocks, tables and subroutines. The appendices include examples of release and recovery program shells.

SECTION I

BACKGROUND

MRP began in the Georgia Strait in 1973 and expanded to cover the entire B.C. coast's commercial and sport programs in 1974. The main directive of the program was and is to collect and process release and recovery information on coded-wire tagged (CWT) salmonids. However, until 1982 inadequate attention to data quality, processing and user access significantly deterred from the usefulness of this data. Commencing in 1982, a Regional initiative was undertaken to alleviate these problems. The role of BSB in this initiative was to design and implement an efficient database, and to develop analytical and report writer programs for this data. An efficient specialized database system has been completed but some of the historical data from U.S. agencies has yet to be entered into this database.

SYSTEM DESIGN

Data on commercial samples are present from 1975, on marked escapement from 1973 and on releases from 1967. Direct data access uses chaining, binary searches, pointers and backward and forward references. Data access is done through high level FORTRAN programs. Some knowledge of the data structure can be used to make the accessing programs run more efficiently, however, detailed knowledge of the data files is not necessary for users to use the subroutines, common blocks and tables.

DATA STRUCTURE

Separate files are maintained by year for catch/sample, escapement/sample and recovery data. For example, for any given year; there is a catch sample file, an escapement file and a recovery file. However, all CWT marked releases are in one file and all non-CWT releases are in another file regardless of release year. To minimize expensive file overhead, programs should attempt to process the data sets by year. The release file can be accessed sequentially or directly by tagcode. Recovery data can be accessed sequentially, or directly by catch region and time period, or by tagcode. Restrictions on the data such as hatchery, species or brood year which are categorized as release information will determine the path into the data. Here the program would be driven by the release files using pointers found in the release record to link into recovery data. However, if restrictions are by recovery fields such as catch region and period of catch, it may be more efficient to use routines that go directly into the recovery data.

REPORTS

Documentation for the MRP system consists of four separate reports. The intent of this report is to provide programming support for user/programmers. Further system information on datafiles and system flowcharts will be available through in-house reports. These are "THE MRP-REPORTER PROGRAM: A DATA EXTRACTION AND REPORTING TOOL FOR THE MARK RECOVERY PROGRAM DATABASE" by B. R. Kuhn which details a menu of easy to use, self-documenting programs; "AN OVERVIEW OF THE MARK RECOVERY SYSTEM IN THE PACIFIC REGION" by M. A. Holmes which documents background and details of the release and recovery files; and in-house documentation on data files and system flowcharts.

UPDATES

System updates are done periodically to incorporate new data or to process revisions to existing data. In order to keep the system as current as possible, these updates are carried out when required. Release data are entered onto the database as information on the release is received. Recovery data is maintained on a yearly basis. The system status report, available via the MRP menu allows users to see what and when these updates were performed.

SECTION II

COMMON BLOCKS

Introduction

The Mark Recovery database system uses three labelled common blocks:

[CWTSYS.RELEASE]RELRECCOM.FOR	-	CWT release master common block
[FINCLIP.RELEASE]FINRELCOM.FOR	-	FINCLIP release master common block

Cwt Release Common Block

DRC1:[CWTSYS.RELEASE]RELRECCOM.FOR

```
C-----------------------------------------------------
C  COMMON BLOCKS FOR A RELEASE RECORD
C
      INTEGER*4 IVAR(25)
      REAL*4    RVAR(05)
      INTEGER*4 YRADDR(10)
      BYTE      YRSBAC(10)
C
C
      COMMON /RELRECNBLK/ IVAR,RVAR,YRADDR,YRSBAC
C
      CHARACTER AGENCY*4,COMMENT*50,STAGECOD*4
      CHARACTER TAG*8,DELSW*1,FLAG1*1,STUDY*1,COMPLT*1
      CHARACTER MANDFLAG*15
C
      COMMON /RELRECCBLK/  TAG,DELSW,FLAG1,STUDY,COMPLT,
     .AGENCY,COMMENT,STAGECOD,MANDFLAG
C
C     IVARS
C
C  01 HART                          !HART species code
C  02 BROODYEAR
C  03 RUN
C  04 DAY FIRST RELEASE
C  05 MON
C  06 YEAR
C  07 DAY LAST REL
C  08 MON
C  09 YR
C     IF DD-MM-YY OF DATE LAST REL ALL 0
C     MANDFLAG(8:8)=Y
C     THEN RELEASE HAD 100% MORTALITY
C  10 NUM TAGGED
C  11 AD_ONLY
C  12 UNCLIPPED
C  13 TOTAL
C  14 DAYS HELD  "-1" =  UNKNOWN.
C  15 SIZE CODE 1- FSH/LB 2- GR/FISH 3-LEN CM 0-NO DATA
C  16 DAY LAST UPDATE
C  17 MON LAST UPDATE
C  18 YR  LAST UPDATE
C  19 HATCHERY NUM (ADDR IN SITES FILE)
C  20 REL SITE "
C  21 STOCK    "
C  22 COORDINATOR CODE
C      ON CHAINS, CHAIN=0  MEANS YOU GOT THE LAST ONE
```

```
C       "CURRENT"
C       *WATCH FOR CHAIN BREAKS*
C  23   CHAIN TO LAST RECORD SAME HAT
C  24     "       "     "      "  RELSITE
C  25     "                       STOCK
C
C
C   RVARS
C  01 %TAGLOSS
C  02 SIZE  SEE IVAR(15)
C  03 BIOSTD (EXPECTED SURVIVAL)  .LT. 0.0 = UNKNOWN
C  SPARES
C
C    OTHER STUFF
C
C    COMPLT =Y  RELEASE IS COMPLETE
C           =N  INCOMPLETE
C    DELSW  =Y  RELEASE IS "DELETED",NOT YET DROPPED VIA
C               MAINT
C               IE - AN INVALID - NOGOOD - IGNORE
C           =N  IS VALID.
C               THIS OPTION ALLOWS SYS ACCESS TO DEAD
C               RECORDS.
C
C     COMPLETION LEVEL.
C     A "Y" IN THE CORRESPONDING POSITION OF MANDFLAG
C     MEANS THAT FIELD IS COMPLETE.(USE TO SEE WHY
C     RELEASE IS INCOMPLETE)
C
C     1AGENCY,2HART,3BY,4STK,5STUDY
C     6HAT,   7REL, 8RDL,9NTAG,10ADPERTL
C    11UNCLIP
C
C     AGENCY - release agency handling the tagging study
C     HART - species code
C     BY - brood year
C     STK - stock of fish used in the study
C     STUDY - production, experimental etc.
C     HAT - hatchery of study
C     REL - release site
C     RDL - last date released (month and year)
C     NTAG - number tagged
C     ADPERTL - adipose, percent tagloss (estimated % of
C                 shed  tags at time of sampling
C     UNCLIP - number of unclipped fish released with  C
      tagged group
C
C     YRSBAC - CONTAINS THE LAST TWO DIGITS OF A YEAR IN
C              WHICH THERE WAS A RETURN.
C     YRADDR - CONTAINS START CHAIN ADDRESS
C              IN THE RECTAGSyy.DAF FILE.
C_____________________________________
```

Finclip Release Common Block

DRC1:[FINCLIP.RELEASE]FINRELCOM.FOR

```
C
C-------------------------------------------------
C  COMMON BLOCKS FOR A FINCLIP RELEASE RECORD
C  FINCLIP RTNS WERE "LIFTED" FROM CWT RTNS. NOTE THAT
C  BECAUSE OF THIS,SOME OF THE FIELDS IN FINCLIP RECORDS
C  ARE NOT MEANINGFUL AS DESCRIBED.
C  ALSO - NOTE THAT THE MAIN DIFFERENCES BETWEEN CWT
C  FIELD NAMES AND FINCLIP FIELD NAMES ARE SO:
C
C      CWT                  FINCLIP
C
C     IVARS                 JVARS
C     RVARS                 SVARS
C     YRADDR,YRSBAC         ZRADDR,ZRSBAC
C     ON CHARACTER FIELDS,THE FIRST CHARACTER OF THE CWT
C     CHARACTER VARIABLE IS REPLACE WITH AN "F" FOR
C     FINCLIP.
C     EG   AGENCY   FGENCY  EXCEPT FLAG1 IS FFLAG1
C
C
      INTEGER*4 JVAR(25)
      REAL*4    SVAR(05)
      INTEGER*4 ZRADDR(10)
      BYTE      ZRSBAC(10)
C
C
      COMMON /FINRELNBLK/ JVAR,SVAR,ZRADDR,ZRSBAC
C
      CHARACTER FGENCY*4,FOMMENT*50,FTAGECOD*4
      CHARACTER FAG*8,FELSW*1,FFLAG1*1,FTUDY*1,FOMPLT*1
      CHARACTER FANDFLAG*15
C
      COMMON /FINRELCBLK/  FAG,FELSW,FFLAG1,FTUDY,FOMPLT,
     .FGENCY,FOMMENT,FTAGECOD,FANDFLAG
C
C     JVARS
C
C  01 HART
C  02 BROODYEAR
C  03 RUN
C  04 DAY FIRST RELEASE
C  05 MON
C  06 YEAR
```

```
C  07 DAY LAST REL
C  08 MON
C  09 YR
C     IF DD-MM-YY OF DATE LAST REL ALL 0 AND
C     FANDFLAG(8:8)=Y
C     THEN RELEASE HAD 100% MORTALITY
C  10 NUM TAGGED
C  11 AD_ONLY
C  12 UNCLIPPED
C  13 TOTAL
C  14 DAYS HELD  "-1" =  UNKNOWN.
C  15 SIZE CODE 1- FSH/LB 2- GR/FISH 3-LEN CM 0-NO DATA
C  16 DAY LAST UPDATE
C  17 MON LAST UPDATE
C  18 YR  LAST UPDATE
C  19 HATCHERY NUM (ADDR IN SITES FILE)
C  20 REL SITE "
C  21 STOCK    "
C  22 COORDINATOR CODE
C      ON CHAINS, CHAIN=0  MEANS YOU GOT THE LAST ONE
C      "CURRENT"
C      *WATCH FOR CHAIN BREAKS*
C  23  CHAIN TO LAST RECORD SAME HAT
C  24    "       "     "      "  RELSITE
C  25    "                      STOCK
C
C
C   SVARS
C  01 %TAGLOSS
C  02 SIZE  SEE JVAR(15)
C  03 BIOSTD (EXPECTED SURVIVAL)  .LT. 0.0 = UNKNOWN
C  SPARES
C
C    OTHER STUFF
C
C    FOMPLT =Y  RELEASE IS COMPLETE
C           =N  INCOMPLETE
C    FELSW  =Y  RELEASE IS "DELETED",NOT YET DROPPED VIA
C               MAINT    IE - AN INVALID - NOGOOD - IGNORE

C           =N  IS VALID.
C               THIS OPTION ALLOWS SYS ACCESS TO DEAD
C               RECORDS.
C
C     COMPLETION LEVEL.
C     A "Y" IN THE CORRESPONDING POSITION OF FANDFLAG
C     MEANS THAT FIELD IS COMPLETE.(USE TO SEE WHY RELEASE
C     INCOMPLETE)
C
C     1AGENCY,2HART,3BY,4STK,5STUDY
C     6HAT,   7REL, 8RDL,9NTAG,10ADPERTL
C    11UNCLIP
```

```
C
C
C     AGENCY - release agency handling the tagging
C              study
C     HART - species code
C     BY - brood year
C     STK - stock of fish used in the study
C     STUDY - production, experimental etc.
C     HAT - hatchery of study
C     REL - release site
C     RDL - last date released (month and year)
C     NTAG - number tagged
C     ADPERTL - adipose, percent tagloss (estimated %
C               of shed tags at time of sampling
C     UNCLIP - number of unclipped fish released with
C              tagged group
C
C     ZRSBAC - CONTAINS THE LAST TWO DIGITS OF A YEAR IN
C              WHICH THERE WAS A RETURN.
C     ZRADDR - CONTAINS START CHAIN ADDRESS
C              IN THE RECTAGSyy.DAF FILE.

C-------------------------------------------------------------
```

Commercial Recovery Common Block

DRC1:[CWTSYS.RECOVERED]RECTAGCOM.FOR

```
C_____________________________________
C       COMMON BLOCK FOR A TAGRECOVERY
C
C       'CWT:[CWTSYS.RECOVERED]RECTAGCOM.FOR'
C
C       RECFORMAT      1 = MARINE RECOVERIES
C                      2 = FRESHWATER RECOVERIES
C
C
C                                          VALID IN RECFORMAT
C
      CHARACTER
     .RECTAG*8,      !TAGCODE                          1 2
     .RECTYPE*1,     !RECOVERY TYPE                    1 2
     .RECGRADE*1,    !GRADE (S-M-L)                    1
     .RECSPLOC*7,    !RECOV LOCATION - SPORT ONLY      1
     .RECAGETYP*1,   !SAMPLE AGE A,J,I,M (EG ADULT)      2
     .RECSITEC*1,    !SITE CODE (H/R)                    2
     .RECSTAT*3      !STAT AREA                        1
C
      COMMON /CWTRECCHARBLK/
     .RECTAG,RECTYPE,RECGRADE,RECSPLOC,RECSTAT,RECSITEC,
     .RECAGETYP
C
C
      BYTE
     .RECBY,         !BROOD YEAR                       1 2
     .RECSEX,        !SEX-MAT                          1
     .RECOLOUR,      !COLOUR 1-R 2-W 3-MIX             1
     .RECWK,         !CATCH FISHING PERIOD             1
     .RECLENC,       !LENGTH CODE 1-FK 2-HYP 3-TOT     1
     .RECWTC,        !WEIGHT CODE 1-RND 2-DRESS        1
     .RECRUN,        !RECOVERY RUN TYPE (0-4)            2
     .RECGEAR        !GEAR CODE                        1 2
C
C
      COMMON /CWTRECBYTBLK/
     .RECBY,RECSEX,RECOLOUR,RECWK,RECLENC,RECWTC,
     .RECRUN,RECGEAR
C
C
```

```
      INTEGER*2
     .RECREGION,     !CATCH REGION                  1 2
     .RECWT,         !WT  99V9                      1
     .RECLEN,        !LEN  MM                       1
     .RECSUBAREA,    !GUESS (COMMERCIAL)            1
     .RECSPECIES     !SPECIES (HART NUM)            1 2
C
      COMMON /CWTRECI2BLK/
     .RECREGION,RECWT,RECLEN,RECSUBAREA,RECSPECIES
C
      INTEGER*4
     .RECNREC,       !NUMBER OF RECOVERIES          1 2
     .RECSITE,       !SITE ADDRESS                    2
     .RECCSPER,      !PERIOD POINTER CATSAM,SPTFAC  1
     .RECCDNSP,      !SPECIES POINTER CATSAM  "     1 2
     .RECFORMAT,     !RECORD FORMAT TYPE            1 2
     .RECCHAIN       !PREVIOUS ADDR - SAME TAG      1 2
C
      COMMON /CWTRECI4BLK/
     .RECNREC,RECSITE,RECCSPER,RECCDNSP,RECFORMAT,
     .RECCHAIN
C
C------------ END INCLUDED RECOVERY RECORD COMMON ------
```

SOURCE CODE MODULES

Introduction

Some routines access data via passed arguments. Descriptions for some of these arguments are available in user source code modules.

Commercial Sample Source Code

DRC1:[CWTSYS.CATSAM]CSRECCOM.FOR

```
C-----------------------------------------
C       DESCRIPTION A CATCH-SAMPLE RECORD
C
        INTEGER*4 CSREC(40,10,2)
C        40 PERIODS  10 SPECIES   SAMPLE,CATCH (PIECES)
C
C        SPECIES POINTERS:
C        SP: 1 - 118 SOX    2 - 124 CHINOOK  3 - 115 COHO
C            4 - 112 DOG    5 - 108 PINK     6 - 128 STLHEAD
C            7 - 126 MASU                    8 -     N/U
C            9 - 127 CUTTHROAT               10  -      N/U
C
C          END  DESCRIPTION A CSREC
C          ----------------------------
```

Escapement Sample Source Code

DRC1:[CWTSYS.ESCAPE]ESRECCOM.FOR

```
C-------------------------------------------
C      DESCRIPTION AND DECLARATION OF AN ESCAPEMENT
C      "CATCH/SAMPLE" RECORD  (VIA CWTESGET)
C
      INTEGER*4 ESREC(2,10,4,6)
C
C     RANDOM/SELECT, SPECIES,   SAM-AGE,    1-SAMPLE
C       1      2             1 = A/ADULT,    2-TOT       C
C                                  RUN(RANDOMS)
C                            2 = J/JACK,     3-SUM NO-PINS
C                            3 = I/IMMATURE 4-SUM LOST-PINS
C                            4 = M/MIXED      (IE LP+CO+BI)
C                                            5-SUM NO-DATAS
C                                            6-SUM "KNOWN"
C                                              TAGS
C
C
C       SPECIES POINTERS:
C       SP: 1 - 118 SOX    2 - 124 CHINOOK   3 - 115 COHO
C           4 - 112 DOG    5 - 108 PINK      6 - 128 STLHEAD
C           7 - 126 MASU   8 - 000 UNKNOWN
C           9 - 127 CUTTHROAT
C          10  -     N/U
C
C         END  DESCRIPTION A ESREC
C         ----------------------------
```

Random Recovery Common Block

DRC1:[CWTSYS.RECOVERED]RECSUMCOM.FOR

```
C-------------------------------------------
C       DESCRIPTION OF RECOVERED TAGS MATRIX
C      *** NOTE USE OF SPECIES "POT" 8. FOR UNKNOWN SPECIES
C      *** THIS IS DIFFERENT THAN THE OTHER FORMS OF THIS
C          MATRIX.
C      *** DOES NOT INCLUDE "FRESHWATER" CATCH
C          REGIONS.(CR.GE.90)
C      *** INCLUDES ONLY RANDOM RECOVERIES.
C          RECOVERY TYPES 0 THRU 9, BLANK, L, D, Z, R
C
        INTEGER*4 SUMREC(54,10,4)
C       GIVEN A CR AND YR     54 PERIODS  10 SPECIES
C
C       1-NP(NO-PIN) 2-LP(LOST-PIN (INCL CO'S (COLOUR) &
C       BI'S(BINARY))
C       3-ND (NO-DATA)  4-KNOWNS
C
C          NOTE: "RE-" (UNREAD RARE EARTH) NOT
C          INCLUDED.ANYWHERE.
C
C        SPECIES POINTERS:
C        SP: 1 - 118 SOX    2 - 124 CHINOOK  3 - 115 COHO
C            4 - 112 DOG    5 - 108 PINK     6 - 128 STLHEAD
C            7 - 126 MASU   8 - 000 UNKNOWN*
C            9 - 127 CUTTHROAT
C           10  -      N/U
C
C          END  DESCRIPTION A NON-SUM
C          ---------------------------
```

Sport Recovery Matrix

DRC1:[CWTSYS.CATSAM]SPTFACREC.FOR

```
C-------------------------------------------------
C
C     DESCRIPTION OF THE SPORT FACTOR MATRIX
C
C     VIA CWTMI2 RTN...
C
      REAL*4 SPTFAC(25:29,12,10,2)
C
C                    CR   MM SP     1 - ESTIMATED MARKS
C                                   2 - OBSERVED
C                                       RECOVERIES *
C      A NEGATIVE (-1.0) IN ESTIMATED MARKS INDICATES
C      SYSTEM HAS NO SAMPLE DATA FOR THIS CR,MM,SP.
C      A ZERO INDICATES A SAMPLE WITH 0 MI
C
C      * - OBSERVED RECOVERIES EXCLUDES NO-PINS
C
C       SPECIES POINTERS:
C
C       SP: 1 - 118 SOX   2 - 124 CHINOOK  3 - 115 COHO
C           4 - 112 DOG   5 - 108 PINK     6 - 128 STLHEAD
C           7 - 126 MASU  8 -    N/U
C           9 - 127 CUTTHROAT
C          10  -     N/U
C
C         END  DESCRIPTION SPORT FACTOR MATRIX
C         ----------------------------
```

Sport Mark Incidence Counts

DRC1:[CWTSYS.CATSAM]CMIREC.FOR

```
C-----------------------------------------
C       DESCRIPTION OF MI MATRIX FOR CDN SPORT
C       MARK INCIDENCE COUNTS. (OBSERVED)
C
        INTEGER*4 MI(25:29,12,10)
C
C        25:29    CDN SPORT CATCH REGIONS..
C        12 PERIODS  (MONTHS)    10 SPECIES
C       A NEGATIVE (-1) IN OBSERVED MARKS INDICATES SYSTEM
C       HAS NO SAMPLE DATA FOR THIS CR,MM,SP.
C       A ZERO INDICATES A SAMPLE WITH 0 MI.
C
C                    USES "CDN" NUMBERS.SEE SUBRTN SPECIES.

C
C        SP: 1 - 118 SOX    2 - 124 CHINOOK   3 - 115 COHO
C            4 - 112 DOG    5 - 108 PINK      6 - 128 STLHEAD
C            7 - 126 MASU   8 -     N/U
C            9 - 127 CUTTHROAT
C           10  -      N/U
C
C          END  DESCRIPTION MARK INCI MAT
```

TABLES

Introduction

Most MRP's tables are carried internally within the subroutines. Data included within these tables are documented briefly in this report (see "AN OVERVIEW OF THE MARK RECOVERY SYSTEM IN THE PACIFIC REGION"). Some tables are loaded from the library on the first call (UNIT 100), and these are documented in detail below the chart of table types and relevant code data.

NOTE : TABLES ARE SUBJECT TO CHANGE.

Table Type Chart

TABLE TYPES	SUBROUTINE NAME	TABLE CODES
Escapement age codes	CwtAjim	AGE CODES A - adult J - jack I - immature M - mature
Gear type codes	CwtCmStCr	GEAR CODES
Gear conversion	CwtGrNam RegionGr	10 - GILLNET 15 - MIXED NET 20 - SEINE NET 30 - MIXED TROLL 31 - FREEZER TROLL 32 - DAY TROLL 33 - ICE TROLL 07 - SPORT

Production area (ProdArea.Tab)	Cwtparea	see PRODAREA TABLE
Program type	CwtProgTyp	PROGRAM TYPES

CDP Community Economic Dev.
PIP Public Involvement Program
RES Research
OPS Enhancement Operations
SPD Special Projects Div.
SPU Small Projects Unit
PRV Province of BC

Rearing type	CwtRearTyp	REARING CODES

H - Hatchery
W - Wild
U - Unknown

Catch region (RegNam.Tab)	CwtRegNam	see REGNAM TABLE
Colour code	CwtColour	COLOUR CODES

Numeric	Colour
72	BL - BLUE
73	BR - BROWN
93	CY - CHROME YELLOW
99	DB - DARK BLUE
95	GD - GOLD
97	GM - METALLIC GREY
75	GN - GREEN
74	GY - GREY
96	LA - LAVENDER
76	LB - LIGHT BLUE
77	LG - LIGHT GREEN
94	MG - MEDIUM GREEN
87	MX - MIXED
78	OR - ORANGE
82	PK - PINK
83	PU - PURPLE
84	RD - RED
98	VI - VIOLET
70	WH - WHITE
80	XR - OXIDE RED

81	XY - OXIDE YELLOW
86	YW - YELLOW
##	## - SOLIDS
$$	$$ - STRIPES

Nontagcodes	CwtNonTag	NONTAGCODES CODES

NO-PIN
NO PIN
R-EARTH
R EARTH
LOST-PIN
LOST PIN
NO-DATA
NO DATA
COLOUR
BINARY

Release Stage	CwtStage	STAGE CODE

EE	Eyed Egg
UF	Unfed Fry
FF	Fed Fry
FFS	Fed Fry Seapen
UFC	Unfed Fry Channels
FFL	Fed Fry Lake
YE	Yearlings
SM	Smolts
SSM	Smolts Seapen

Sexual Maturity	CwtSexMat	MATURITY CODES

1 = Immature Female
2 = Mature Female
3 = Immature Male
4 = Mature Male
5 = Unknown

Grade	CwtSml	GRADE CODES

S - Small
M - Medium
L - Large
U - Unknown

Stock Type	CwtStkTyp	STOCK CODES H - Hatchery W - Wild M - Mixed U - Unknown
Sports statarea/ catch region (SpStatCr.Tab)	CwtSpStCr	See SPORT CATCH REGION TABLE
Agency microcode	MicroTran	MICROTRAN TABLE

1st 2 characters = agency code
next 4 characters = agency
next 3 characters = country
last 2 characters = coordinator code

```
01WDF USA04
02CDFOCDN03
03NMFSUSA
04ADFGUSA
05FWS USA07
06CDFGUSA08
07ODFWUSA05
08CDFRCDN03
09ODFWUSA05
10IDFGUSA10
11UW  USA04
12BCFWCDN09
13WDF USA04
14FWS USA07
15WDF USA04
21NIFCUSA14
23NMFSUSA
31ADFGUSA
47MIC USA13
60OAF USA05
63WDF USA04
B0UAJ USA01
H1WDF USA04
H2CDFOCDN03
H3NMFSUSA
H4ADFGUSA
H5FWS USA07
H6CDFGUSA08
H7ODFWUSA05
$$NMFSUSA
```

```
##NMFSUSA
HFADFGUSA
XXOAF USA05
XROAF USA05
ADFGUSA
ANADUSA05
COOPUSA04
ELWAUSA07
HOH USA07
LUMMUSA07
NISQUSA07
SQAXUSA07
STILUSA07
SUQ USA07
TULAUSA07
YAKIUSA07
DOMSUSA04
NSRAUSA01
QDNRUSA
OSU USA05
SOF USA08
SJ  USA01
SKAGUSA07
WDG USA
UAJ USA
BHSRUSA05
MUCKUSA07
SSC USA14
SSRAUSA01
OPSRUSA
PPWRUSA
PUYAUSA14
DIPCUSA
PNPTUSA
SKOKUSA
CIAAUSA12
```

Run type name	RunType	RUNTYPE TABLE
		1 = SPRING
		2 = SUMMER
		3 = FALL
		4 = WINTER
		5 = HYBRID
		6 = LANDLOCKED

Sample location	SamLoc	SAMPLE LOCATIONS TABLE 01 = STEVESTON 02 = VANCOUVER 03 = VICTORIA 04 = NAMU 05 = PRINCE RUPERT 06 = PORT HARDY 07 = TOFINO 08 = UCLUELET 09 = HISTORICAL 10 = HISTORICAL 11 = HISTORICAL 12 = GEORGIA STRAIT 14 = WINTER HARBOUR 15 = MASSETT 16 = CAMPBELL RIVER 17 = COURTENAY/COMOX 18 = NANAIMO/FRENCH CREEK
Species code	Species	HART SPECIES TABLE 108 = PINK 112 = CHUM 115 = COHO 118 = SOCKEYE 124 = CHINOOK 126 = MASU 127 = CUTTHROAT 128 = STEELHEAD
Catch region/statarea /subarea	Statsub	see STATSUB.DAT in [CWTSYS.TABLES]
Study type	StudyType	STUDY TYPE TABLE W = WILD P = PRODUCTION M = MONITORING B = BOTH PRODUCTION & MONITORING E = EXPERIMENTAL

TABLE DESCRIPTIONS

Production Area Table

Source File Name : PRODAREA.TAB
Disk:[Directory] : CWT:[CWTSYS.TABLES]

Purpose of Table :
This table is used for looking up prodarea names given the prodarea code.

Format	Element Description
A100	1st 4 characters are the production area code - remaining fields are production area name

Element Relationships or Furthur Description

CODE NAME

NASS- NASS RIVER - NASS RIVER & TRIBS (STAT 3)
SKNA- SKEENA RIVER - SKEENA RIVER & TRIBS (STAT 4)
QCI - QUEEN CHARLOTTE ISLANDS - STAT AREA 2
NCST- NORTH COAST - STATS 3 - 5 EXCEPT SKNA & NASS
CCST- CENTRAL COAST - STATS 6 - 8
RIVR- RIVERS/SMITH INLETS - STATS 9 - 10
NWVI- NORTHWEST VANCOUVER ISLAND - STATS 25 - 27
SWVI- SOUTHWEST VANCOUVER ISLAND - STATS 20 - 24
JNST- JOHNSTONE STRAIT - STATS 11 - 13
GSVI- GEORGIA STRAIT VANCOUVER ISLAND - STATS 14, 17, 18, 19
GSML- GEORGIA STRAIT MAINLAND - STATS 15, 16, 28, 29
LWFR- LOWER FRASER RIVER - FRASER RIVER & TRIBS BELOW HOPE
THOM- THOMPSON RIVER - THOMPSON RIVER & TRIBS
UPFR- UPPER FRASER RIVER - FRASER RIVER & TRIBS ABOVE HOPE
KOOT- KOOTENAYS
YUKN- YUKON
WA01- WASH. MANAGEMENT AREA 1
WA02- WASH. MANAGEMENT AREA 2
WA03- WASH. MANAGEMENT AREA 3
WA04- WASH. MANAGEMENT AREA 4
WA05- WASH. MANAGEMENT AREA 5
WA06- WASH. MANAGEMENT AREA 6
UPWA- UPPER WASHINGTON

LWWA- LOWER WASHINGTON
SNAK- SNAKE RIVER
HEAD- HEAD WATERS (COLUMBIA)
BRGT- BRIGHTS (COLUMBIA)
LOCO- LOWER COLUMBIA
WILL- WILLIAMETTE
DESC- DESCHUTES
UPOR- UPPER OREGON
LWOR- LOWER OREGON
WEAK- WESTERN ALASKA
NOAK- NORTHERN OUTSIDE ALASKA
NIAK- NORTHERN INSIDE ALASKA
SOAK- SOUTHERN OUTSIDE ALASKA
SIAK- SOUTHERN INSIDE ALASKA
CALI- CALIFORNIA
SACR- SACRAMENTO
UNKN- TERRA INCOGNITA

Catch Region Name Table

Table Source File Name : REGNAM.TAB
Disk:[Directory] : CWT:[CWTSYS.TABLES]

Purpose of Table :
This table is used for looking up catch region abbreviated names given the catch region number.

Table Element	Format	Element Description
FLAG(200)	Byte	Set to 1 if catch region I is read in.
NAMES(200)	A20	The abbreviated catch region names.

Element Relationships or Furthur Description

CATCH REGION CODE		CATCH REGION NAME (STAT AREAS)
0001	NWTR	NorthWest Vancouver Is. Troll 25 - 27)
0002	SWTR	SouthWest Vancouver Is. Troll (21, 23, 24)
0003	WOT	Washington Oregon Troll (historical)
0004	GSTR	Georgia Strait Troll (13 - 18, 29)
0005	CTR	Central Troll (historical, now 56 & 57)
0006	NTR	Northern Troll (1 - 5)
0007	ATR	Alaska Troll (historical)
0008	FGN	Fraser Gillnet (29)
0009	NN	Northern Net (1 - 5)
0010	GSN	Georgia Strait Net (14 - 18)
0011	JSN	Johnstone Strait Net (11, 12)
0012	CN	Central Net
0013	JFN	Juan de Fuca Net (20)
0014	JFTR	Juan de Fuca Troll
0015	NWTR & CTR	NorthWest Vancouver Is. & Central Troll
0017	NWTR & SWTR	NorthWest & SouthWest Vancouver Is. Troll
0018	NTR &CTR	Northern and Central Troll
0019	JSN & CN	Johnstone Strait and Central Net

0020	NWVN	NorthWest Vancouver Is. Net (25, 26, 27)
0021	SWVN	SouthWest Vancouver Is. Net (21 - 24)
0025	NSPT	Northern Sport (1 - 5)
0026	CSPT	Central Sport (6 - 12, 30)
0027	WSPT	Westcoast Sport (21, 23 - 27)
0028	GSPT	Georgia Strait Sport (13 - 20, 28, 29)
0029	Freshwater Sport	
0033	NN & CN	Northern & Central Net
0034	GSTR & CTR	Georgia Strait and Central Troll
0036	YKN	Yukon Net
0037	JFN & GSN	Juan de Fuca & Georgia Strait Net
0045	JSN & GSN	Johnstone Strait & Georgia Strait Net
0046	FGN & GSN	Fraser Gillnet & Georgia Strait Net
0047	AN	Alaska Net (historical)
0048	BC	British Columbia
0053	GSTR & SWTR	Georgia Strait & SW Vancouver Is. Troll
0056	NCTR	North Central Troll (6 - 9, 30)
0057	SCTR	South Central Troll (10 - 12)

Sport Catch Region Table

Table Source File Name : SPSTATCR.TAB
Disk:[Directory] : CWT:[CWTSYS.TABLES]

Purpose of Table :
This table is used to find out catch region for statistical area and check that the catch region is a sport region.

Table Element	Format	Element Description
STAB(100)	A3	Array of statistical area codes.
CTAB(100)	I4	Array of corresponding catch region numbers.

Element Relationships or Furthur Description

STATISTICAL AREA	SPORT CATCH REGION
STAT AREA= 001	CR=25
STAT AREA= 003	CR=25
STAT AREA= 004	CR=25
STAT AREA= 005	CR=25
STAT AREA= 006	CR=26
STAT AREA= 007	CR=26
STAT AREA= 008	CR=26
STAT AREA= 009	CR=26
STAT AREA= 010	CR=26
STAT AREA= 011	CR=26
STAT AREA= 012	CR=26
STAT AREA= 013	CR=28
STAT AREA= 014	CR=28
STAT AREA= 015	CR=28
STAT AREA= 016	CR=28
STAT AREA= 017	CR=28

```
STAT AREA= 018     CR=28
STAT AREA= 019     CR=28
STAT AREA= 020     CR=28
STAT AREA= 021     CR=27
STAT AREA= 023     CR=27
STAT AREA= 024     CR=27
STAT AREA= 025     CR=27
STAT AREA= 026     CR=27
STAT AREA= 027     CR=27
STAT AREA= 028     CR=28
STAT AREA= 029     CR=28
STAT AREA= 02E     CR=25
STAT AREA= 02W     CR=25
STAT AREA= 030     CR=26
STAT AREA= 0FW     CR=29
STAT AREA= 29B     CR=28
STAT AREA= 29C     CR=28
STAT AREA= 29D     CR=28
STAT AREA= 29E     CR=28
```

SUBROUTINE OVERVIEW

Introduction

Varied and detailed subroutines have been written in order that programmers/users can access the database without extensive knowledge of the systems files and structure. Below is a short explanation of the subroutines used. Following that is the subroutine hierarchy where all subroutines are listed and those subroutines calling other routines are shown. This list is followed by those subroutines using limited unit numbers reserved for the system. User programs must not use units 60 through 100 as they are reserved for current and future system usage. A detailed description of each subroutine, its purpose, arguments used etc. is then documented.

Subroutines

CWTAJIM - Handles catch/sample age pointers for escapement.
CWTCALWEK - Calculates sport recovery date.
CWTCHAIN - Performs site chaining through the release master file.
CWTCMSTCR - Given statarea code and gear code; returns the catch region.
CWTCOLOUR - Converts numeric tagcode to alpha colour tagcode and vice versa.
CWTCRWK - Given recovery year, catch region, and fishing period; returns the start record and number of records to read in the recovered tagcodes file.
CWTCSGET - Given the recovery year and catch region; returns the catch sample record (40 fishing periods, 10 species and catch/sample number.
CWTESGET - Given site, runtype and year; returns escapement catch/sample record.
CWTGRNAM - Given gear code returns gear name.
CWTHISITE - Returns highest valid site number.
CWTMI1 - Given year; returns Canadian mark incidence counts for all sport catch regions (25:29).
CWTMI2 - Given recovery year; returns estimated marks and observed recoveries for the sport catch.

CWTMMW - Converts dates to fishing periods.
CWTNONGET - Given the recovery year and tagcode will return he starting record number in the recovered tagcodes file.
CWTNONOPN - Used by CWTNONGET to open address files.
CWTNONTAG - Verifies valid non-tagcodes and returns species code.
CWTOPCHN - Opens sites chain index file for accessing the finclip or CWT release master file by site.
CWTOPMST - Opens release master and returns number of records in file.
CWTPAREA - Given prodarea code, returns number of valid codes on file and the prodarea name.
CWTPROGTYP - Given program code (not SEP) will return program name and number of table entries.
CWTREARTYP - Given rearing type code returns rearing name and number of table entries.
CWTRECRD - Reads recovery records.
CWTREGNAM - Given catch region number, returns catch region name and vice versa.
CWTRELGET - Reads the release master data file (tagcode access).
CWTRELRD - Reads from release master and stores it in the common block (sequential access).
CWTSEXMAT - Determines sex maturity code.
CWTSITGET - Given the site number and option code; returns the site name, province/state code and prodarea code.
CWTSML - Given the weight, dress code, and gear code; returns fish grade for chinooks.
CWTSPSTCR - Given statarea code; returns sport catch region.
CWTSTAGE - Given the release stage code: returns the release stage name.
CWTSTKTYP - Converts stock type codes.
CWTSUM1 - Adjusts for NO-DATA's and LOST-PINS.
FINOPMST - Opens finclip release master and returns number of records.
FINRELGET - Reads the finclip release master file (release ID access).

FINRELRD - Reads RECNUM from release master and stores it in the common block (sequential access).

MICROTRAN - Given the releasing agency micro-code; returns the agency name code, coordinator code and country code. Given agency, returns country.

REGIONGR - Validates a catch region/gear code combination.

RELSTAGE - Returns release stage name given stage code or vice versa.

RUNTYPE - Given the run number; returns the run name and vice versa.

SAMLOC - Verifies sample location numbers.

SPECIES - Given the Hart species code; returns the name and vice versa.

STATSUB - Verifies catch region, statarea, sub area combinations.

STUDYTYPE - Verifies the given study type code and returns the study type name.

Subroutine Hierarchy

All subroutines are listed here. Those subroutines which access or call other subroutines will show those subroutines.

```
CwtAjim
CwtCalWek

CwtChain
  CwtOpMst
  CwtOpChn
  CwtRelReader

CwtCmStCr
  RegionGr

CwtColour
CwtCrWk
CwtCsGet
CwtEsGet
CwtGrNam
CwtHiSite
CwtMi1
CwtMi2
CwtMmw

CwtNonGet
  CwtNonOpn

CwtNonOpn
CwtNonTag
CwtOpChn
CwtOpMst
CwtParea
CwtProgTyp
CwtRearTyp

CwtRecRd
  Taglook
  CwtSpec1
  CwtNonTag

CwtRegNam

CwtRelGet
  CwtOpMst
  LoadTheHints (internal routine only)
  CwtRelReader
```

```
CwtRelRd (must call Cwtopmst first)
CwtSexMat
CwtSitGet
CwtSml
CwtSpStCr
CwtStage
CwtStkTyp
CwtSuml
FinOpMst

FinRelGet
  FinOpMst
  LoadFinHints
  FinRelReader

FinRelRd (must be opened by Finopmst first)

MicroTran
RegionGr
RelStage
RunType
SamLoc
Species
StatSub
StudyType
```

Subroutines and Restricted Numbers

Subroutine	Unit #'s used
CWTCHAIN	90, 99
CWTCMSTCR	100
CWTCRWK	94
CWTCSGET	98
CWTESGET	83
CWTMI1	100
CWTMI2	100
CWTNONGET	81
CWTNONOPN	81
CWTOPCHN	99
CWTOPMST	90
CWTPAREA	100
CWTRECRD	97, 92, 93
CWTREGNAM	100
CWTRELGET	90, 95
CWTRELRD	90
CWTSITGET	96, 97
CWTSPSTCR	100
FINOPMST	70
FINRELGET	70, 75
FINRELRD	70
STATSUB	91

```
CWTAJIM

Source File Name : CWTAJIM.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN


Purpose of Subroutine :
    Handles catch/sample age pointers for escapement
    recoveries.  If character (CAJIM) is passed, number
    (IAJIM) and name (CNAME) are returned.  If number
    (IAJIM) is passed, character (CAJIM)  and name
    (CNAME) are returned.
    See ESRECCOM.FOR  for valid codes.


Arguments Passed :
    CAJIM     Character*1  ----  Alpha character for
                                   escapement age.
    IAJIM     Integer*4    ----  Numeric escapement age.
    CNAME     Character*10 ----  The escapement age.
    OPT       Integer*4    ----  The option code.

Arguments Returned :
    CAJIM     Character*1  ----  The alpha character for
                                  escapement age.
    IAJIM     Integer*4    ----  The numeric escapement
                                  age.
    CNAME     Character*10 ----  The escapement age.
    ERROR     Integer*4    ----  The error code.


Subroutine Call Example :
    Call Cwtajim( CAJIM,IAJIM,CNAME,OPT,ERROR )

Access Restrictions :
    NONE

Option Codes / Description :
    OPT = 1  ----  Given alpha age code, returns
                   numeric age and description.
    OPT = 2  ----  Given numeric age code, returns
                   alpha code and description.

Error Condition Codes / Description :
    ERR = 0  ----  No errors.
    ERR = 1  ----  Errors

Called From :
    Source File Name : yourfil.ext
    Disk:[Directory] : your:[directory]
```

CWTCALWEK

Source File Name : CWTCALWEK.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine calculates the sport recovery date. If day (D), month (M), or year (Y) is passed, week (W) is returned. If month (M), year (Y) and week (W) are passed, day (D) is returned.

Arguments Passed :

D	----	The day.
M	----	The month.
Y	----	The year.
W	----	The calender week.
OPT	----	The option code.

Arguments Returned :

D	----	The day.
M	----	The month.
Y	----	The year.
W	----	The calender week.
ERROR	----	The error code.

Subroutine Call Example :

```
Call Cwtcalwek( D,M,Y,W,OPT,ERR )
```

Access Restrictions :
NONE

Option Codes / Description :

OPT = 1	----	Given day, month and year; calender week is returned.
OPT = 2	----	Given month, year and week, last day of week is returned.

Error Condition Codes / Description :
    ERR = 0 ---- No errors.
    ERR = 1 ---- Error in date.
    ERR = 2 ---- Option error.

Called From :
    Source File Name : yourfil.ext
    Disk:[Directory] : your:[directory]

Special Considerations :
    1. Sport data are only expandable at the month level, though sport recoveries may show an actual recovery week.
    2. Calender weeks are not the same as statistical weeks (statweeks). They do not cross month boundaries and may be less than a 7 day period. Week 1 ends the first Saturday of the month. Last week ends the last day of the month.

CWTCHAIN

Source File Name : CWTCHAIN.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine retrieves records from the release master file (ADDR) by hatchery (OPT=1), release site (OPT=2) and stock (OPT=3) and puts it into the release master common blocks. Records from the CWT release file (JOPT=0) or finclip release file (JOPT=1) can be retrieved in reverse tagcode sequence.

Arguments Passed :

ADDR	Integer*4	----	Record number in the sites file used to find the record in the release master file.
OPT	Integer*4	----	Option code.
JOPT	Integer*4	----	Option code.

Arguments Returned :

ADDR	Integer*4	----	Record number in the release file.
ERR	Integer*4	----	Error result code.

Subroutine Call Example :

```
Call CwtChain( ADDR,OPT,JOPT,ERR )
```

Access Restrictions :
NONE

Option Codes / Description :

OPT = 1	----	Return last record same hatchery.
OPT = 2	----	Return last record same release site.
OPT = 3	----	Return last record same stock.
JOPT = 0	----	From CWT release master file.
JOPT = 1	----	From FINCLIP release master file.

Error Condition Codes / Description :

```
    ERR = 0  ---- Okay.
    ERR = 1  ---- Invalid address ( ADDR must be between 1
                  and 9999).
    ERR = 2  ---- 'END OF FILE', no record read.
    ERR = 4  ---- Chain break: change has been made to the
                  release master file and the maintenance
                  program has not been run to restructure
                  the chains. Record in common block may
                  not have been updated.
                  Contact MRP technician.
```

Included Common Blocks :

```
    Internal Block Names : RELRECNBLK/RELRECCBLK
    Source File Name     : RELRECCOM.FOR
    Disk:[Directory]     : CWT:[CWTSYS.RELEASE]
```

Called From :

```
    Source File Name : yourfil.ext
    Disk:[Directory] : your:[directory]
```

Other Subroutines Called :

```
    Source File Name    : CWTOPMST.FOR( NDUM )
    Source File Name    : CWTOPCHN.FOR (start pointer)
    Source File Name    : CWTRELRD.FOR ( RECNUM )
    Disk:[Directory]    : CWT:[CWTSYS.TABLES]
```

Special Considerations :

1. This routine uses Fortran file units 90, 99 (internal routines only).
2. A change in option (OPT) or address (ADDR), does a "reset".
3. JOPT is applied only on 1st call.

CWTCMSTCR

```
Source File Name : CWTCMSTCR.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
    For commercial recoveries; statistical area (STAT) and
    gear (GR) are passed and catch region (CR) is returned.


Arguments Passed :
    STAT      Character*3  ----  The statistical area
                                 (statarea) code.
    GR        Integer*4    ----  The numeric gear code.
    OPT       Integer*4    ----  The option code.

Arguments Returned :
    CR        Integer*4    ----  Catch region returned.
    ERR       Integer*4    ----  The error code.

Subroutine Call example :
    Call CwtCmStCr( CR,STAT,GR,OPT,ERR )

Access Restrictions :
    NONE

Option Codes / Description :
    OPT = 1  ----  Statarea and gear are passed; catch
                   region is returned.

Error Condition Codes / Description :
    ERR = 0  ----  No errors.
    ERR = 1  ----  Unknown statarea.
    ERR = 2  ----  Gear code invalid.
    ERR = 3  ----  Catch region for this statarea/gear code
                   does not exist.
    ERR = 99 ----  A given statarea/gear code combination
                   points to more than one catch region.
                   Writes error log to FOR100.DAT.
                   Call SSA support group.

Called From :
    Source File Name : yourfil.ext
    Disk:[Directory] : your:[directory]

Other Subroutines Called :
    Source File Name : REGIONGR.FOR
    Disk:[Directory] : CWT:[CWTSYS.TABLES]
```

Other Inputs :
    Source File Name : STATSUB.DAT
    Disk:[Directory] : CWT:[CWTSYS.TABLES]

Other Outputs :
    File Name : FOR100.DAT (see error code 99).

Special Considerations :
    1. (GR) argument must contain new gear codes.
    2. Uses Fortran file unit 100 and leaves closed (internal routine only).
    3. On the first call, the program builds lookup tables from the STATSUB.DAT file using the subroutine REGIONGR and uses the tables to test for 'non-unique' statarea/gear code combinations (see error code 99).
    4. STATSUB.DAT is used for verifying data entry records.

CWTCOLOUR

Source File Name : CWTCOLOUR.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine or Program :
Converts numeric colour tags (NTAG) to alpha colour tags (ATAG) and vice versa.

Arguments Passed :

NTAG	Character*8	----	The numeric colour tagcode.
ATAG	Character*8	----	The alpha colour tagcode.
OPT	Integer*4	----	The option code.

Arguments Returned :

NTAG	Character*8	----	The numeric colour tagcode.
ATAG	Character*8	----	The alpha colour tagcode.
ERR	Integer*4	----	The error result code.

Subroutine Call Example :
Call CwtColour(NTAG,ATAG,OPT,ERR).

Option Codes / Description :
OPT = 1 ---- Numeric tag passed, alpha tag returned.
OPT = 2 ---- Alpha tag passed, numeric tag returned.

Error Condition Codes / Description :
ERR = 0 ---- No errors.
ERR = 1 ---- Conversion error.

Called From :
Source File Name : yourfil.ext
Disk:[Directory] : your:[directory]

Special Considerations :
1. Translation stops if/when a "*" suffix is found (* indicates repeated uses of the same code).
2. ## (solids) and $$ (stripes) are valid tag prefixes.

CWTCRWK

Source File Name : CWTCRWK.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Given a recovery year (YR), catch region (CR), and fishing period (IPR); this routine gives starting address START) and number of records to read (NREC) for tag recovery records in recovery file RECTAGSyy.DAF.

Arguments Passed :

YR	Integer*4	----	The recovery year.
CR	Integer*4	----	The numeric catch region code.
IPR	Integer*4	----	The fishing period.
OPT	Integer*4	----	The option code.

Arguments Returned :

START	Integer*4	----	Starting record number in tagcode file.
NREC	Integer*4	----	The number of records to read.
ERR	Integer*4	----	The error result code.

Subroutine Call Example :
Call CwtCrWk(YR,CR,IPR,START,NREC,OPT,ERR)

Access Restrictions :
NONE

Option Codes / Description :
OPT = 1 ---- Given recovery year, catch region, and fishing period, returns starting address and number of records.

Error Condition Codes / Description :

ERR = 0 ---- No errors.
ERR = 1 ---- No data for this recovery year.
ERR = 2 ---- Catch region is not valid.
ERR = 3 ---- Fishing period is not valid.

Called From :

Source File Name : yourfil.ext
Disk:[Directory] : your:[directory]

Special Considerations :

1. This routine uses Fortran unit number 94 (internal routine only).
2. Sample code : DO I=START, START+NREC-1.

```
CWTCSGET


Source File Name : CWTCSGET.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
    Gets a catch/sample record (CSREC) for catch region
    (CR) and  recovery year (YR) combination.

Arguments Passed :
    CR        Integer*4  ----  The numeric catch region
                               code.
    YR        Integer*4  ----  The recovery year .
    OPT       Integer*4  ----  The option code.

Arguments Returned :
    CSREC     Integer*4(40,10,2)  ----  The matrix of catch
                                        sample ratios.
    ERR       Integer*4           ----  The error result
                                        code.

Subroutine Call Example :
    Call CwtCsGet( CR,YR,CSREC,OPT,ERR )

Access Restrictions :
    NONE

Option Codes / Description :
    OPT = 1  ----  Given recovery year; catch region
                   returns the catch/sample record.

Error Condition Codes / Description :
    ERR = 0  ----  No errors.
    ERR = 1  ----  No data for this recovery year.
    ERR = 2  ----  No data for this catch region.

Included Source Code :
    Source File Name : CSRECCOM.FOR
    Disk:[Directory] : CWT:[CWTSYS.RECOVERED]


Called From :
    Source File Name : yourfil.ext
    Disk:[Directory] : your:[directory]

Special Considerations :
    1. This routine uses Fortran unit number 98 (internal
       routine only).
```

CWTESGET

Source File Name : CWTESGET.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine gets an escapement catch/sample record (ESREC) for site (ISITE), runtype (RUN) and year (YR).

Arguments Passed :

ISITE	Integer*4	----	The numeric site address.
CHR	Character*1	----	Either H (hatchery) or R (relsite).
RUN	Integer*4	----	The runtype (1 - 6).
YR	Integer*4	----	The year of recovery.
OPT	Integer*4	----	The option code.

Arguments Returned :

ESREC	Integer*4(2,10,4,6)	----	The escapement catch sample matrix.
ERR	Integer*4	----	The error code.

Subroutine Call Example :
Call Cwtesget(ISITE,CHR,RUN,YR,ESREC,OPT,ERR)

Access Restrictions :
NONE

Option Codes / Description :
OPT = 1 ---- Given site address, hatchery (H) or release site (R), run and year; get the escapement catch sample record.

Error Condition Codes / Description :
ERR = 0 ---- No errors.
ERR = 1 ---- No data for this year.
Err = 2 ---- No data for this site and runtype.

Included Source Code :
    Source File Name : ESRECCOM.FOR
    Disk:[Directory] : DRC1:[CWTSYS.ESCAPE]

Called From :
    Source File Name : yourfil.ext
    Disk:[Directory] : your:[directory]

Special Considerations :
    1. This routine uses Fortran unit number 83 (internal routine only).

```
CWTGRNAM


Source File Name : CWTGRNAM.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN


Purpose of Subroutine :
    Given either old or new gear codes (GR); gear name
    (NAME) is returned.


Arguments Passed :
    GR      Integer*4      ----  Old or new numeric gear
                                 code.
    OPT     Integer*4      ----  The option code.

Arguments Returned :
    NAME    Character*15   ----  Alpha gear name.
    ERR     Integer*4      ----  The error result code.

Subroutine Call Example :
    Call CwtGrNam( GR,NAME,OPT,ERR )

Access Restrictions :
    NONE

Option Codes / Description :
    OPT = 1  ----  Old gear code passed;  gear name
                   returned.
    OPT = 2  ----  New gear code passed; gear name
                   returned.
    OPT = 3  ----  Old gear code passed; new gear code and
                   name returned.

Error Condition Codes / Description :
    ERR = 0  ----  No errors.
    ERR = 1  ----  Gear code does not exist.
    ERR = 2  ----  Option requested is not valid.


Called From :
    Source File Name : yourfil.ext
    Disk:[Directory] : your:[directory]
```

CWTHISITE

Source File Name : CWTHISITE.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Will return highest valid site numbers for each of release site, hatchery and stock.

Arguments Passed :

OPT	Integer*4	----	The option code.

Arguments Returned :

HIHAT	Integer*4	----	The highest hatchery code.
HIREL	Integer*4	----	The highest release site code.
HISTK	Integer*4	----	The highest stock code.
ERR	Integer*4	----	The option code.

Subroutine Call Example :
```
Call CwtHiSite( HIHAT,HIREL,HISTK,OPT,ERR )
```

Access Restrictions :
NONE

Option Codes / Description :

OPT = 1	----	Returns current highest valid site numbers.

Error Condition Codes / Description :

ERR = 0	----	No errors.
ERR = 1	----	Report to SSA support group.

Called From :
Source File Name : yourfil.ext
Disk:[Directory] : your:[directory]

Special Considerations :
1. Uses unit 100, leaves it closed (internal routine only).
2. For freshwater recoveries, release site may also be recovery site. The site numbers can then also be used for freshwater recovery, catch and sample data.

```
CWTMI1


Source File Name : CWTMI1.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN


Purpose of Subroutine :
    This routine gets the Canadian mark incidence counts
    for all  Canadian sport catch regions.  Pass the year
    (YR) and the  array (MI) is returned.

Arguments Passed :
    YR    Integer*4  ----  The recovery year.
    OPT   Integer*4  ----  The option code.

Arguments Returned :
    MI    Intger*4(25:29,12,10)  ----  Mark incidence
                                       matrix
          (sport catch region, months, species).
    ERR   Integer*4                ----  The error code.

Subroutine Call Example :
    Call Cwtmi1( YR,MI,OPT,ERR )

Access Restrictions :
    NONE

Option Codes / Description :
    OPT = 1  ----  Pass the recovery year and mark
                   incidence array  is returned.

Error Condition Codes / Description :
    ERR = 0  ----  No errors.
    ERR = 1  ----  No recovery data for this year.
    ERR = 2  ----  Not a valid option.

Included Source Code :
    Source File Name : CMIREC.FOR
    Disk:[Directory] : CWT:[CWTSYS.CATSAM]
```

Called From :
    Source File Name : yourfil.ext
    Disk:[Directory] : your:[directory]

Special Considerations :
    1. This routine uses Fortran unit 100 (internal unit only).
    2. One call per recovery year will return all data.
    3. A -1 in the observed marks indicates system has no sample data for this catch region, month or species.
    4. A 0 indicates the sample has no mark incidence.

CWTMI2

Source File Name : CWTMI2.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine gets Canadian estimated marks in the sport catch along with observed recoveries (SPTFAC) when passed the recovery year (YR).

Arguments Passed :

YR	Integer*4	----	The recovery year.
OPT	Integer*4	----	The option code.

Arguments Returned :

SPTFAC	Real*4(25:29,12,10,2)	---- The sport recovery matrix (catch region, months, species, observed & estimated records).
ERR	Integer*4	---- The error code.

Subroutine Call Example :

```
Call CWTMI2( YR,SPTFAC,OPT,ERR )
```

Access Restrictions :
NONE

Option Codes / Description :

OPT = 1	----	Pass the recovery year and sport recovery is returned.

Error Condition Codes / Description :

ERR = 0	----	No errors.
ERR = 1	----	No recovery data for this year.
ERR = 2	----	Not a valid option.

Included Source Code :
    Source File Name : SPTFACREC.FOR
    Disk:[Directory] : CWT:[CWTSYS.CATSAM]

Called From :
    Source File Name : yourfil.ext
    Disk:[Directory] : your:[directory]

Special Considerations :
    1. This routine uses Fortran unit 100 (internal unit only) and leaves it closed.
    2. One call per recovery year will return all data.
    3. A -1 in the observed marks indicates system has no sample data for this catch region, month or species.
    4. A 0 indicates the sample has no mark incidence.
    5. Observed recoveries excludes NO-PINS.

CWTMMW

Source File Name : CWTMMW.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
For Canadian commercial statweeks, converts day (D), month (M), year (Y) to catch month (MM), week (W) and fishing period (PERIOD); or it takes catch month (MM), week (W) and returns fishing period (PERIOD); or converts fishing period (PERIOD), to month (MM) and week (W); or converts year (Y), month (MM) and week (W) to day (D) and month (MM).

Arguments Passed :

D	Integer*4	---- The day of the month.
M	Integer*4	---- The month of the year.
Y	Integer*4	---- The year.
MM	Integer*4	---- The catch month.
W	Integer*4	---- The catch week.
PERIOD	Integer*4	---- The fishing period.
OPT	Integer*4	---- The option code.

Arguments Returned :

MM	Integer*4	---- The catch month.
W	Integer*4	---- The catch week.
PERIOD	Integer*4	---- The fishing period.
ERR	Integer*4	---- The error result code.

Subroutine Call Example :

```
Call CwtMMW( D,M,Y,MM,W,PERIOD,OPT,ERR )
```

Access Restrictions :
NONE

Option Codes / Description :

OPT = 1	----	Day, month, year passed; catch month, week, fishing period returned.
OPT = 2	----	Catch month and week passed; fishing period returned.
OPT = 3	----	Fishing period passed; catch month and week returned.
OPT = 4	----	Year, catch month and week passed; day and month returned. (Returns data for the ending period).

```
Error Condition Codes / Description :
    ERR = 0  ----  No errors.
    ERR = 1  ----  Input date error.
    ERR = 2  ----  From OPT = 2,4  catch month and week is
                   not valid.
    ERR = 3  ----  From OPT = 3, fishing period is not
                   valid (1-39).

Called From :
    Source File Name : yourfil.ext
    Disk:[Directory] : your:[directory]
```

CWTNONGET

Source File Name : CWTNONGET.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Given recovery year (YR) and non-tagcode (TAG); returns the starting address (RECNUM) for non-tagcodes (No-Pin, Lost-Pin, No-Data etc.) from the recovery files RECTAGSyy.DAF's.

Arguments Passed :

TAG	Character*8	----	Non-tagcode.
YR	Integer*4	----	Recovery year required.
OPT	Integer*4	----	The option code.

Arguments Returned :

RECNUM	Integer*4	----	Beginning record number for non-tagcodes.
ERR	Integer*4	----	The error result code.

Subroutine Call Example :

```
Call CwtNonGet( TAG,YR,RECNUM,OPT,ERR )
```

Access Restrictions :
NONE

Option Codes / Description :
OPT = 1 ---- Given tagcode and recovery year; return the starting address.

Error Condition Codes / Description :
ERR = 0 ---- No errors.
ERR = 1 ---- Tagcode not found.
ERR = 2 ---- No tagcodes for this recovery year.

Called From :
Source File Name : yourfil.ext
Disk:[Directory] : your:[directory]

Other Subroutines Called :
    Source File Name : CWTNONOPN.FOR
    Disk:[Directory] : CWT:[CWTSYS.TABLES]

Special Considerations :
    1. On the first call for the year, all addresses are loaded and saved.
       Handles up to 50 years per run before there is a table overflow.
    2. This routine uses Fortran unit number 81 (internal routine only).

CWTNONOPN

Source File Name : CWTNONOPN.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Given recovery year (YR) this routine will open the non-tagcode starting chain address file for indexing to the RECTAGSyy.DAF file of recovered tagcodes.

Arguments Passed :

YR	Integer*4	----	The recovery year required.
OPT	Integer*4	----	The option code.

Arguments Returned :

ERR	Integer*4	----	The error result code.

Subroutine Call Example :

```
Call CwtNonOpn( YR,OPT,ERR )
```

Access Restrictions :
NONE

Option Codes / Description :

OPT = 1	----	Opens the file recovery file.

Error Condition Codes / Description :

ERR = 0	----	No errors.
ERR = 1	----	No tags available for this recovery year.

Called From :
Source File Name : CWTNONGET.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]

Special Considerations :
1. This routine uses Fortran unit number 81 (internal routine only).

CWTNONTAG

Source File Name : CWTNONTAG.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
If the passed tagcode (TAG) is a NO-PIN etc., it and the passed species code (SPECIES) are combined to create a new tagcode i.e. NP-115. Another option takes the passed tagcode (TAG) and validates it as a valid non-tagcode (TAG); then species (SPECIES) is returned.

Arguments Passed :

TAG	Character*8	----	Non-tagcode.
SPECIES	Integer*4	----	Species combined with non-tagcode.
OPT	Integer*4	----	The option code.

Arguments Returned :

TAG	Character*8	----	Valid tagcode/species combination.
SPECIES	Integer*4	----	The species code retrieved from the tagcode.
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call CwtNonTag( TAG,SPECIES,OPT,ERR )
```

Access Restrictions :
NONE

Option Codes / Description :

OPT = 1	----	Non-tagcode and species are passed, and a valid non-tagcode is returned.
OPT = 2	----	A valid non-tagcode is passed, and species is returned.
OPT = 3	----	Verifies that the passed tagcode is a valid non-tagcode.

Error Condition Codes / Description :

ERR = 0	----	No error.
ERR = 1	----	Tagcode is not a valid non-tagcode.
ERR = 2	----	Species passed is an invalid species code.

Called From :
Source File Name : yourfil.ext
Disk:[Directory] : your:[directory]

CWTOPCHN

Source File Name : CWTOPCHN.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine opens the sites chaining index file for CWT or Finclip files.

Arguments Passed :
OPT Integer*4 ---- The option code.

Arguments Returned :
NONE

Subroutine Call Example :
```
Call CwtOpChn (OPT)
```

Access Restrictions :
NONE

Option Codes / Description :
OPT = 0 ---- Opens the CWT sites chaining index file.

OPT = -1 ---- Opens the Finclip sites chaining index file.

Error Condition Codes / Description :
NONE

Called From :
Source File Name : CWTCHAIN.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]

Special Considerations :
1. This routine uses Fortran unit number 99 (internal routine only).
2. OPT = -1 was used in some old programs to open finmark release file.
3. See CWTCHAIN.FOR.

CWTOPMST

Source File Name : CWTOPMST.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine opens the release master file and returns the number of records (NREC) in the file .

Arguments Passed :
NONE

Arguments Returned :

NREC	Integer*4	----	The number of release records in the file.

Subroutine Call Example :

```
Call CwtOpMst( NREC )
```

Access Restrictions :
NONE

Error Condition Codes / Description :
NONE

Called From :
Source File Name : yourfil.ext
Disk:[Directory] : your:[directory]

Special Considerations :

1. Record 1 has a null tag code and contains the number of releases (NREC).
2. If NREC is passed as -1 then [FINCLIP.RELEASE]RELEASE.DAF is opened (this option is used only in some old programs for compatibility).
   NOTE : The finmark subroutine FINOPMST should be used.
3. This routine uses Fortran unit number 90 (internal routine only).
4. Sample code : DO I=Z, NREC.

CWTPAREA

Source File Name : CWTPAREA.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Given the production area (prodarea) code (PROD); returns the number of valid prodareas on file (NPROD), the full name of the prodarea (PNAME), the length of the name (PNAMLEN), and the position in the table (NADD); or given the position in the table will return PROD, NPROD, PNAME and PNAMLEN.

Arguments Passed :

PROD	Character*4	----	The alpha prodarea code.
NADD	Integer*4	----	The numeric table position of a prodarea.
OPT	Integer*4	----	The option code.

Arguments Returned :

NPROD	Integer*4	----	Number of prodarea codes in the table.
PROD	Character*4	----	The alpha prodarea code.
PNAME	Character*80	----	The prodarea name.
PNAMLEN	Integer*4	----	The length of the prodarea name.
NADD	Integer*4	----	Numeric position of a prodarea in the table.
ERR	Integer*4	----	The error result code.

Subroutine Call Example :

```
Call CwtParea( NPROD,PROD,PNAME,PNAMLEN,NADD,OPT,ERR )
```

Access Restrictions :
NONE

Option Codes / Description :

OPT = 1	----	Given prodarea code will return prodarea information.
OPT = 2	----	Given position in the table; will return prodarea information.

```
Error Condition Codes / Description :
    ERR = 0    ----  No error.
    ERR = 1    ----  The prodarea code passed is
                     unrecognized.
    ERR = 2    ----  The table passed is outside of the
                     table range.
    ERR = 99   ----  "Table Overflow" occurred, call SSA
                     support group.


Called From :
    Source File Name : yourfil.ext
    Disk:[Directory] : your:[directory]

Other Inputs :
    Source File Name : PRODAREA.TAB
    Disk:[Directory] : CWT:[CWTSYS.TABLES]


Special Considerations :
    1. This routine uses Fortran unit number 100 on the
       first call then saves the table array in an internal
       file and closes the unit off (internal routine only.
```

CWTPROGTYP

Source File Name : CWTPROGTYP.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
When program code (PROG) passed, program name (PROGNAME) and number of table entries (NVAR) is returned; or when table entry number passed (NVAR), program code (PROG) and program name (PROGNAME) returned.

Arguments Passed :

PROG	Character*3	----	The program code.
PROGNAME	Character*40	----	The program name.
NVAR	Integer*4	----	The table entry number.
OPT	Integer*4	----	The option code.

Arguments Returned :

PROG	Character*3	----	The program code.
PROGNAME	Character*40	----	The program name.
NVAR	Integer*4	----	The table entry number.
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call CwtProgTyp( PROG,PROGNAME,NVAR,OPT,ERR)
```

Access Restrictions :
NONE

Option Codes / Description :

OPT = 1	----	Program code passed, program name and number of entries returned.
OPT = 2	----	Nothing passed, # of entries returned.
OPT = 3	----	Entry number passed, program code and name returned.

```
Error Condition Codes / Description :
    ERR = 0  ----  No errors.
    ERR = 1  ----  Program code not in table (from OPT 1).
    ERR = 1  ----  Entry number (NVAR) not in table (from
                   OPT 3).
    ERR = 2  ----  Invalid option code.
    ERR = 3  ----  Internal error.  NOTIFY SSA Support
                   Group.


Called From :
    Source File Name : yourfil.ext
    Disk:[Directory] : your:[directory]

Other Inputs :
    Source File Name : PROGTYP.TAB
    Disk:[Directory] : CWT:[CWTSYS.TABLES]

Special Considerations :
    1. This routine uses Fortran unit number 100 (internal
       routine only).
```

CWTREARTYP

Source File Name : CWTREARTYP.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
If release rearing type (CODE) is passed, returns rearing name (NAME) and number of table entries (NVAR); or returns number of entries in table (NVAR); or if passed table entry number (NVAR), will return rearing type (CODE) and rearing name (NAME).

Arguments Passed :

CODE	Character*1	----	The rearing code.
NAME	Character*20	----	The rearing name.
NVAR	Integer*4	----	The table entry number.
OPT	Integer*4	----	The option code.

Arguments Returned :

CODE	Character*1	----	The rearing code.
NAME	Character*20	----	The rearing name.
NVAR	Integer*4	----	The table entry number.
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call CwtRearTyp( CODE,NAME,NVAR,OPT,ERR )
```

Access Restrictions :
NONE

Option Codes / Description :

OPT = 1	----	Return rearing type name and NVAR given rearing code.
OPT = 2	----	Returns maximum table entries.
OPT = 3	----	Returns rearing code and name given table entry.

Error Condition Codes / Description :

ERR = 0	----	No errors.
ERR = 1	----	Invalid code passed (option 1).
ERR = 2	----	Invalid NVAR passed (option 3).

Called From :
Source File Name : yourfil.ext
Disk:[Directory] : your:[directory]

CWTRECRD

Source File Name : CWTRECRD.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine reads in the record specified by RECNUM from the recovered tagcode data file RECTAGSyy.DAF and places the record in the recovered tagcodes common blocks. The variable RECCHAIN in the common block is updated with the next record in reverse tagcode sequence. If RECCHAIN is 0 then the last record is in common block.

Arguments Passed :

YR	Integer*4	----	The recovery year.
RECNUM	Integer*4	----	The record number in the file.
OPT	Integer*4	----	The option code.

Arguments Returned :

ERR	Integer*4	----	The error result code.

Subroutine Call Example :

```
Call CwtRecRd( YR,RECNUM,OPT,ERR )
```

Access Restrictions :
NONE

Option Codes / Description :

OPT = 1	----	Read the record (RECNUM) from the recovered tagcode file.

Error Condition Codes / Description :

ERR = 0	----	No error.
ERR = 1	----	No recovery file for specified recovery year (YR).
ERR = 2	----	Read error - probably invalid RECNUM.
ERR = 3	----	Record found but the release is no longer in the current release master data file. Species number or brood year may be in error. Contact MRP technician.

Included Common Blocks :

Internal File Name :CWTRECCHARBLK/
CWTRECBYTBLK/CWTRECI2BLK/CWTRECI4BLK

Source File Name : RECTAGCOM.FOR
Disk:[Directory] : CWT:[CWTSYS.RECOVERED]

Called From :
Source File Name : yourfil.ext
Disk:[Directory] : your:[directory]

Other Subroutines Called :
Source File Name : TAGLOOK.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]

Source File Name : CWTNONTAG.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]

Source File Name : CWTSPEC1.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]

Special Considerations :

1. If tag is a non-tagcode then brood year is set to 0 in the recovery record common block.
2. This routine uses Fortran unit number 97. The routine TAGLOOK called by this routine use Fortran unit numbers 92 & 93 (internal routines only).
3. Returns the sport location data when present.

CWTREGNAM

Source File Name : CWTREGNAM.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine will return the catch region name (NAME) given the region number (REG); or will return the catch region number (REG) given the catch region name (NAME). The highest catch region number (HIVAL) is always returned.

Arguments Passed :

REG	Integer*4	----	The numeric catch region code.
NAME	Character*20	----	Character name of the catch region.
OPT	Integer*4	----	The option code.

Arguments Return :

REG	Integer*4	----	Numeric catch region code.
NAME	Character*20	----	Character name of the catch region.
HIVAL	Integer*4	----	Highest catch region numeric code.
ERR	Integer*4	----	The error result code.

Subroutine Call Example :

```
Call CwtRegNam( REG,NAME,HIVAL,OPT,ERR )
```

Access Restrictions :
NONE

Option Codes / Description :

OPT = 1	----	Return catch region name given the catch region number.
OPT = 2	----	Return catch region number given catch region name.

```
Error Condition Codes / Description :
    ERR = 0  ----  No error.
    ERR = 1  ----  Catch region is not in the table.


Called From :
    Source File Name : yourfil.ext
    Disk:[Directory] : your:[directory]

Other Inputs :
    Source File Name : REGNAM.TAB
    Disk:[Directory] : CWT:[CWTSYS.TABLES]

Special Considerations :
    1. This routine uses Fortran unit number 100 on first
       call then  closes it (internal routine only).
```

CWTRELGET

Source File Name : CWTRELGET.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Does a direct read of the release master data file using the passed tagcode (TAGCODE); or a read of the release master reading the record following the previous record read.

Arguments Passed :

TAGCODE	Character*8	----	The tagcode.
OPT	Integer*4	----	The option code.

Arguments Returned :

ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call CwtRelGet( TAGCODE,OPT,ERR )
```

Access Restrictions :
NONE

Option Codes / Description :

OPT = 1	----	Given tagcode; read corresponding release record and put into release master common blocks.
OPT = 2	----	Read record following previous record read. (File is in tagcode sequence).

Error Condition Codes / Description :

ERR = 0	----	No error.
ERR = 1	----	Tagcode not found.
ERR = 2	----	End of file, no record read (from option 2).

```
Included Common Blocks :
    Internal File Name : RELRECNBLK/RELRECCBLK
    Source File Name   : RELRECCOM.FOR
    Disk:[Directory]   : CWT:[CWTSYS.RELEASE]


Called From :
    Source File Name : yourfil.ext
    Disk:[Directory] : your:[directory]


Other Subroutines Called :
    Source File Name : CWTOPMST.FOR
    Source File Name : CWTRELGET.FOR
    Source File Name : CWTRELRD.FOR
    Disk:[Directory] : CWT:[CWTSYS.TABLES]


Special Considerations :
    1. This routine uses Fortran unit numbers 90 & 95
       (internal  routine only).
```

CWTRELRD

Source File Name : CWTRELRD.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine reads record (RECNUM) from the release master data file and stores it in the release master common blocks.

Arguments Passed :
RECNUM Integer*4 ---- Record number.

Arguments Returned :
NONE

Subroutine Call Example :
Call CwtRelRd(RECNUM)

Access Restrictions :
NONE

Option Codes / Description :
NONE

Error Condition Codes / Description :
NONE

Included Common Blocks :
Internal Name : RELRECNBLK/RELRECCBLK

Source File Name : RELRECCOM.FOR
Disk:[Directory] : CWT:[CWTSYS.RELEASE]

Called From :
Source File Name : yourfil.ext
Disk:[Directory] : your:[directory]

Other Inputs :
Source File Name : RELEASE.DAF
Disk:[Directory] : CWT:[CWTSYS.RELEASE]

Special Considerations :
1. This routine uses Fortran unit number 90 (internal routine only).
2. File must be opened by the subroutine CWTOPMST.

```
CWTSEXMAT


Source File Name : CWTSEXMAT.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN


Purpose of Subroutine :
    This routine returns sex (SEX) and maturity (MAT) if
    passed code (CODE); or if passed sex (SEX) and maturity
    (MAT) will return code (CODE).


Arguments Passed :
    SEX     Character*1     ----    The sex code.
    MAT     Character*1     ----    The maturity code.
    CODE    Integer*4       ----    The sex maturity code.
    OPT     Integer*4       ----    The option code.

Arguments Returned :
    SEX     Character*1     ----    The sex code.
    MAT     Character*1     ----    The maturity code.
    CODE    Integer*4       ----    The sex maturity code.
    ERR     Integer*4       ----    The error code.

Subroutine Call Example :
    Call CwtSexMat( CODE,SEX,MAT,OPT,ERR )

Access Restrictions :
    NONE

Option Codes / Description :
    OPT = 1  ----  Given code; returns sex and maturity.
    OPT = 2  ----  Given sex and maturity; returns code.

Error Condition Codes / Description :
    ERR = 0  ----  No errors.
    ERR = 1  ----  Invalid arguments passed.


Called From :
    Source File Name : yourfil.ext
    Disk:[Directory] : your:[directory]
```

CWTSITGET

Source File Name : CWTSITGET.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
For a given hatchery or stock or release area code (ADDR) depending on the option code (COD), this routine returns names where:
STR1 contains the name depending on option code.
STR2 contains the production area code.
STR3 contains the province/state code.

Arguments Passed :

ADDR	Integer*4	----	The site numeric code.
COD	Character*1	----	The site letter i.e. H = Hatchery, R = Release site, S = Stock site.

Arguments Returned :

ERR	Integer*4	----	The error code.
STR1	Character*20	----	The site name associated with the site code.
STR2	Character*4	----	The prodarea code if the site code is R, H and S.
STR3	Character*2	----	The province/state code associated with site.

Subroutine Call Example :

```
Call CwtSitGet( ADDR,COD,ERR,STR1,STR2,STR3 )
```

Access Restrictions :
NONE

Option Codes / Description :

COD = H	----	Return information associated with hatchery number.
COD = R	----	Return information associated with release site.
COD = S	----	Return information associated with stock number.

Error Condition Codes / Description :

ERR = 0	----	No error.
ERR = 1	----	Site is not within the valid range of addresses (ADDR is not between 0 and 9999).

Called From :

Source File Name : yourfil.ext
Disk:[Directory] : your:[directory]

Special Considerations :

1. This routine uses Fortran unit numbers 77 and 96 (internal routines only).

CWTSML

Source File Name : CWTSML.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Given the weight (WT), gear code (GEAR), and dress code (DRESS); this routine will return the grade (GRADE) of the fish as 'S', 'M', 'L', or 'U'. For chinook only.

Arguments Passed :

WT	Real*4	----	Fish weight.
DRESS	Integer*4	----	Dress code of a fish.
GEAR	Integer*4	----	The numeric gear code.
OPT	Integer*4	----	The option code.

Arguments Returned :

GRADE	Character*1	----	The character grade code.
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call CwtSML( WT,DRESS,GEAR,GRADE,OPT,ERR )
```

Access Restrictions :
NONE

Option Codes / Description :
OPT = 1 ---- Pass the weight, gear and dress code, grade code returned.

Error Condition Codes / Description :
ERR = 0 ---- No error.

Called From :
Source File Name : yourfil.ext
Disk:[Directory] : your:[directory]

CWTSPSTCR

Source File Name : CWTSPSTCR.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Given statarea (STAT); returns sports catch region (CR) or verifies that the catch region (CR) is a sport catch region.

Arguments Passed :

CR	Integer*4	----	The numeric sports catch region.
STAT	Character*3	----	The alpha statistical area code.
OPT	Integer*4	----	The option code.

Arguments Returned :

CR	Integer*4	----	The numeric sports catch region.
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call CwtSpStCr( CR,STAT,OPT,ERR )
```

Access Restrictions :
NONE

Option Codes / Description :

OPT = 1	----	Statarea passed, Canadian sports catch region returned.
OPT = 2	----	Catch region passed, verify it as a sports catch region.

Error Condition Codes / Description :

ERR = 0	----	No error.
ERR = 1	----	Can not find catch region for this statarea.
ERR = 2	----	For option 2, catch region is not a sport catch region.

Called From :
    Source File Name : yourfil.ext
    Disk:[Directory] : your:[directory]

Other Inputs :
    Source File Name : SPSTATCR.TAB
    Disk:[Directory] : CWT:[CWTSYS.TABLES]

Special Considerations :
    1. This routine uses Fortran unit 100 and closes it
       after first call (internal routine only).

CWTSTAGE

Source File Name : CWTSTAGE.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This subroutine verifies release stage (STAGECOD) and returns stage name (STAGENAM) and number of table entries (NENT); or if passed the number of table entry (NENT) will return release stage name (STAGENAM) and release stage code (STAGECOD).

Arguments Passed :

STAGECOD	Character*4	----	Alpha release stage code.
STAGENAM	Character*20	----	Release stage name.
NENT	Integer*4	----	Number of table entries.
OPT	Integer*4	----	The option code.

Arguments Returned :

STAGECOD	Character*4	----	Alpha release stage code.
STAGENAM	Character*20	----	Release stage name.
NENT	Integer*4	----	Number of table entries.
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call Cwtstage( STAGECOD,STAGENAM,NENT,OPT,ERR)
```

Access Restrictions :
NONE

Option Codes / Description :

OPT = 1	----	Given release code, returns stage name and entry number.
OPT = 2	----	Given entry number, returns release stage code and name.

```
Error Condition Codes / Description :
    ERR = 0  ----  No errors.
    ERR = 1  ----  Release name not in table (NENT
                   returned).
    ERR = 2  ----  From option = 2, invalid entry
                   number.
    ERR = 3  ----  Invalid option.


Called From :
    Source File Name : yourfil.ext
    Disk:[Directory] : your:[directory]
```

CWTSTKTYP

Source File Name : CWTSTKTYP.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
If release stock type (CODE) is passed, returns name (NAME) and number of entries in table (NVAR); or returns number of entries in table (NVAR); or if passed number of entries in table (NVAR) will return stock type (CODE) and stock name (NAME).

Arguments Passed :

CODE	Character*1	----	The stock code.
NAME	Character*20	----	The stock name.
NVAR	Integer*4	----	The table entry number.
OPT	Integer*4	----	The option code.

Arguments Returned :

CODE	Character*1	----	The stock code.
NAME	Character*20	----	The stock name.
NVAR	Integer*4	----	The table entry number.
ERR	Integer*4	----	The error code.

Subroutine Call Example :
```
Call CwtStkTyp( CODE,NAME,NVAR,OPT,ERR )
```

Access Restrictions :
NONE

Option Codes / Description :

OPT = 1	----	Return stock type name and NVAR given stock code.
OPT = 2	----	Returns maximum table entries.
OPT = 3	----	Returns stock code and name given table entry.

Error Condition Codes / Description :

ERR = 0	----	No errors.
ERR = 1	----	Invalid code passed (option 1).
ERR = 2	----	Invalid NVAR passed (option 3).

Called From :
Source File Name : yourfil.ext
Disk:[Directory] : your:[directory]

CWTSUM1

Source File Name : CWTSUM1.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine is used to adjust a known recovery to reflect NO-DATA and LOST-PIN'S. Pass year (YR) and catch region (CR) to get random marine recoveries (SUMREC).

Arguments Passed :

YR	Integer*4	----	The recovery year.
CR	Integer*4	----	The catch region.
OPT	Integer*4	----	The option code.

Arguments Returned :

SUMREC	Integer*4(54,10,4)	----	The matrix of random recovery data.
LUN	Integer*4	----	Reserved for I/O unit.
ERROR	Integer*4	----	The error code.

Subroutine Call Example :
Call Cwtsum1(YR,CR,SUMREC,LUN,OPT,ERR)

Access Restrictions :
NONE

Option Codes / Description :

OPT = 1	----	Sumrec matrix is not refreshed if routine is called with same year and catch region as previous call.
OPT = 2	----	Matrix is refreshed if same year and catch region as previous call.

Error Condition Codes / Description :

ERR = 0	----	No errors.
ERR = 1	----	No data for this catch region.
ERR = 2	----	No data for this recovery year.

Included Source Code :
Source File Name : RECSUMCOM.FOR
Disk:[Directory] : CWT:[CWTSYS.RECOVERED]

Called From :
Source File Name : yourfil.ext
Disk:[Directory] : your:[directory]

Special Considerations :

1. Option 1 cuts down on I/O in some applications.
2. User must reserve an I/O unit in LUN. User must not use this unit and LUN must not conflict with reserved CWT units.

FINOPMST

Source File Name : FINOPMST.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine opens the finclip release master file RELEASE.DAF and returns the number of records (NREC) in the file .

Arguments Passed :
NONE

Arguments Returned :

NREC	Integer*4	----	The number of release records in the file.

Subroutine Call Example :
Call Finopmst(NREC)

Access Restrictions :
NONE

Error Condition Codes / Description :
NONE

Called From :
Source File Name : yourfil.ext
Disk:[Directory] : your:[directory]

Special Considerations :
1. Record 1 has a null tag code and contains the number of records on file (NREC). See CWTOPMST.
2. This routine uses Fortran unit number 70 (internal routine only).

FINRELGET

Source File Name : FINRELGET.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine or Program :
Does a direct read of the finclip release master data file using the passed tagcode (TAGCODE); or a read of the finclip release master reading the record following the previous record read.

Arguments Passed :

TAGCODE	Character*8	----	The tagcode.
OPT	Integer*4	----	The option code.

Arguments Returned :

ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call Finrelget( TAGCODE,OPT,ERR )
```

Access Restrictions :
NONE

Option Codes / Description :

OPT = 1	----	Given tagcode (TAGCODE), reads corresponding release record and puts the record into release master common blocks.
OPT = 2	----	Read record following previous record read. (File is in tagcode sequence).

Error Condition Codes / Description :

ERR = 0	----	No error.
ERR = 1	----	Tagcode not found.
ERR = 2	----	End of file, no record read (from option 2).

```
Included Common Blocks :
    Internal Source Name : FINRELNBLK/FINRELCBLK
    Source File Name     : FINRELCOM.FOR
    Disk:[Directory]     : CWT:[FINCLIP.RELEASE]


Called From :
    Source File Name : yourfil.ext
    Disk:[Directory] : your:[directory]

Other Subroutines Called :
    Source File Name : FINOPMST.FOR
    Source File Name : FINRELRD.FOR
    Disk:[Directory] : CWT:[CWTSYS.TABLES]

Special Considerations :
    1. This routine uses Fortran unit numbers 70 & 75
       (internal routine only).
    2. Term "TAGCODE" refers to release identifier.
```

FINRELRD

Source File Name : FINRELRD.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine or Program :
This routine reads record from the release master data file and puts it into the release master common blocks.

Arguments Passed :

RECNUM	Integer*4	----	Record number of the release master file.

Arguments Returned :
NONE

Subroutine Call Example :
```
Call Finrelrd( RECNUM )
```

Access Restrictions :
NONE

Option Codes / Description :
NONE

Error Condition Codes / Description :
NONE

Included Common Blocks :
Internal Source Name : FINRELNBLK/FINRELCBLK
Source File Name : FINRELCOM.FOR
Disk:[Directory] : CWT:[FINCLIP.RELEASE]

Called From :
Source File Name : yourfil.ext
Disk:[Directory] : your:[directory]

Special Considerations :
1. This routine uses Fortran unit number 70 (internal routine only).
2. File must be opened by the subroutine FINOPMST before calling FINRELRD.

MICROTRAN

Source File Name : MICROTRAN.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Given agency micro-code (MU); returns the unique agency name code (AGENCY), coordinator code (COORD), and the country code (CNDUSA): or given agency name code (AGENCY); returns the country code (CNDUSA).

Arguments Passed :

MU	Character*2	----	The agency alpha code (1st 2 characters of tagcode).
AGENCY	Character*4	----	The agency alpha name.
OPT	Integer*4	----	The option code.

Arguments Returned :

COORD	Integer*4	----	The coordinator numeric code.
CNDUSA	Character*3	----	The country (CND or USA) alpha code.
ERR	Integer*4	----	The error code.

Subroutine Call Example :
```
Call Microtran( MU,AGENCY,COORD,CNDUSA,OPT,ERR )
```

Access Restrictions :
NONE

Option Codes / Description :

OPT = 1	----	Given agency micro-code; returns the agency name code, coordinator code, and the country code.
OPT = 2	----	Given agency name code; returns the country code.

Error Condition Codes / Description :

ERR = 0	----	No error.
ERR = 1	----	The passed agency micro-code is not defined.
ERR = 2	----	The passed agency name code is not known.

Called From :
    Source File Name : yourfil.ext
    Disk:[Directory] : your:[directory]

Special Considerations :
    1. Some agency codes can be used by more that one agency (i.e. 62 is used by multi-American agencies).

```
REGIONGR


Source File Name : REGIONGR.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
    This routine validates a catch region (REG)/gear code
    (GR) combination.

Arguments Passed :
    REG     Integer*4  ----  The numeric catch region code.
    GR      Integer*4  ----  The numeric gear code.
    OPT     Integer*4  ----  The option code.

Arguments Returned :
    ERR     Integer*4  ----  The error result code.

Subroutine Call Example :
    Call RegionGr( REG,GR,OPT,ERR )

Access Restrictions :
    NONE

Option Codes / Description :
    OPT = 1  ----  Old gear code passed.
    OPT = 2  ----  New gear code passed.

Error Condition Codes / Description :
    ERR = 0  ----  No error.
    ERR = 1  ----  Invalid catch region.
    ERR = 2  ----  Valid gear code not allowed for this
                   catch region.
    ERR = 3  ----  Invalid option code.
    ERR = 4  ----  Invalid gear code.


Called From :
    Source File Name : yourfil.ext
    Disk:[Directory] : your:[directory]
```

```
RELSTAGE


Source File Name : RELSTAGE.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
    Returns the release stage name (STAGENAME) given stage
    code (STAGE) or vice versa.

Arguments Passed :
    STAGE        Character*4    ----  The release stage alpha
                                      code.
    STAGENAME    Character*15   ----  The release stage name.
    OPT          Integer*4      ----  The option code.

Arguments Returned :
    STAGE        Character*4    ----  The release stage alpha
                                      code.
    STAGENAME    Character*15   ----  The release stage name.
    ERR          Integer*4      ----  The error code.

Subroutine Call Example :
    Call RelStage( STAGE,STAGENAME,OPT,ERR )

Access Restrictions :
    NONE

Option Codes / Description :
    OPT = 1  ----  Stage code passed, stage name returned.
    OPT = 2  ----  Stage name passed, stage code returned.

Error Condition Codes / Description :
    ERR = 0  ----  No error.
    ERR = 1  ----  Stage name or stage code not found in
                   the table.
    ERR = 2  ----  Stage name argument is too small to hold
                   the stage code name, needs 15
characters.
    ERR = 3  ----  Stage code argument too small to hold
                   stage code, needs 4 characters.
    ERR = 4  ----  Invalid option code.

Called From :
    Source File Name : yourfil.ext
    Disk:[Directory] : your:[directory]
```

```
RUNTYPE


Source File Name : RUNTYPE.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN


Purpose of Subroutine :
    Returns the run name (NAME) given the run number (RUN)
    or returns the run number (RUN) given the run name
    (NAME).


Arguments Passed or Received :
    RUN     Integer*4     ----  The run number.
    NAME    Character*10  ----  The run name.
    OPT     Integer*4     ----  The option code.


Arguments Passed or Received :
    RUN     Integer*4     ----  The run number.
    NAME    Character*10  ----  The run name.
    ERR     Integer*4     ----  The error CODE.

Subroutine Call Example :
    Call RunType( RUN,NAME,OPT,ERR )

Access Restrictions :
    NONE

Option Codes / Description :
    OPT = 1  ----  Given the run number, returns the run
                   name.
    OPT = 2  ----  Given the run name, returns the run
                   number.

Error Condition Codes / Description :
    ERR = 0  ----  No error.
    ERR = 1  ----  Invalid run name or run number.
    ERR = 2  ----  Invalid option.


Called From :
    Source File Name : yourfil.ext
    Disk:[Directory] : your:[directory]
```

SAMLOC

Source File Name : SAMLOC.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine will verify the given sample location number (LOCN); or return the sample location name (NAME) given the location number (LOCN); or return the sample location number (LOCN) given the location name (NAME). Canadian data only.

Arguments Passed :

LOCN	Integer*4	----	The sample location number.
NAME	Character*26	----	The sample location name.
OPT	Integer*4	----	The option code.

Arguments Returned :

ERR	Integer*4	----	The error result code.

Subroutine Call Example :
Call SamLoc(LOCN,NAME,OPT,ERR)

Access Restrict ons :
NONE

Option Codes / Description :

OPT = 1	----	Verify the location number.
OPT = 2	----	Given the location number, return the location name.
OPT = 3	----	Given the location name, return the location number.

Error Condition Codes / Description :

ERR = 0	----	No error.
ERR = 1	----	Invalid location number.
ERR = 2	----	Invalid location name.

Called From :
Source File Name : yourfil.ext
Disk:[Directory] : your:[directory]

SPECIES

Source File Name : SPECIES.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Given the species Hart code (Hart) or the species Canadian code (CDNCODE); returns the species name (NAME) or vice versa.

Arguments Passed :

Hart	Integer*4	---- The numeric species Hart code.
CDNCODE	Integer*4	---- The numeric Canadian species code.
NAME	Character*10	---- The species name.
OPT	Integer*4	---- The option code.

Arguments Returned :

Hart	Integer*4	---- The numeric species Hart code.
CDNCODE	Integer*4	---- The numeric Canadian species code.
NAME	Character*10	---- The species name.
ERR	Integer*4	---- The error result code.

Subroutine Call Example :
Call Species(Hart,CDNCODE,NAME,OPT,ERR)

Access Restrictions :
NONE

Option Codes / Description :

OPT = 1	---- Given Hart code or Canadian code, return species name and the numeric codes. The Canadian code must be passed zero if Hart code to be used for lookup.
OPT = 2	---- Given the species name, return the Hart code and Canadian code.

```
Error Condition Codes / Description :
    ERR = 0  ----  No error.
    ERR = 1  ----  Invalid species Hart code or Canadian
                   code or name.
    ERR = 2  ----  Invalid option (OPT).


Called From :
    Source File Name : yourfil.ext
    Disk:[Directory] : your:[directory]
```

```
STATSUB


Source File Name : STATSUB.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
    This routine will verify catch region (CR); or verify
    statarea (STAT); or verify subarea (SUB); or verify
    catch region (CR)/statarea (STAT) combination; or
    verify statarea (STAT)/subarea (SUB) combination; or
    return the name (NAME) of the catch region
(CR)/statarea (STAT)/subarea (SUB) combination; or
will verify the catch region
    (CR)/statarea (STAT)/subarea (SUB) combination.

Arguments Passed :
    CR        Integer*4      ----  The numeric catch region
                                   code.
    STAT      Character*3    ----  The statarea code
                                   (alpha/numeric).
    SUB       Integer*4      ----  The numeric subarea code.
    NAMLEN    Integer*4      ----  The length of the above
                                   name.
    OPT       Integer*4      ----  The option code.

Arguments Returned :
    NAME      Character*60   ----  Description of the
                                   stat-subarea.
    ERR       Integer*4      ----  The error code.

Subroutine Call Example :
    Call StatSub( CR,STAT,SUB,NAME,NAMLEN,OPT,ERR )

Access Restrictions :
    NONE

Option Codes / Description :
    OPT = 1  ----  Verifies the given catch region.
    OPT = 2  ----  Verifies the given statarea.
    OPT = 3  ----  Verifies the given subarea.
    OPT = 4  ----  Verifies the given catch
                   region/statarea.
    OPT = 5  ----  Verifies the given statarea/subarea.
    OPT = 6  ----  Returns name of the catch
                   region/statarea/subarea.
    OPT = 7  ----  Verifies the catch
                   region/statarea/subarea.
```

Error Condition Codes / Description :

ERR = 0	----	No errors.
ERR = 1	----	Invalid catch region.
ERR = 2	----	Invalid statarea.
ERR = 3	----	Invalid subarea.
ERR = 4	----	Invalid catch region/statarea combination.
ERR = 5	----	Invalid statarea/subarea combination.
ERR = 6	----	Cannot find catch region/statarea/subarea.
ERR = 7	----	Name argument not long enough, actual name truncated.
ERR = 8	----	Invalid option.

Called From :

Source File Name : yourfil.ext
Disk:[Directory] : your:[directory]

Special Considerations :

1. This routine uses Fortran unit number 91 (internal routine only).
2. If user program not linked to latest version of the table file. First call to statsub will abort program with message for user to relink.
3. For option 7, if subarea is 0 then verify catch region/statarea. If subarea is 0 and statarea is blank or '000' then verify the catch region.

STUDYTYPE

Source File Name : STUDYTYPE.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Verifies the given study type code (CODE) and returns the study name (NAME).

Arguments Passed :

CODE	Character*1	----	The study type alpha code.
OPT	Integer*4	----	The option code.

Arguments Returned :

NAME	Character*20	----	The study type name.
ERR	Integer*4	----	The error code.

Subroutine Call Example :
Call StudyType(CODE,NAME,OPT,ERR)

Access Restrictions :
NONE

Option Codes / Description :
OPT = 1 ---- Given study type code, return the study type name.

Error Condition Codes / Description :
ERR = 0 ---- No error.
ERR = 1 ---- Invalid study type code.

Called From :
Source File Name : yourfil.ext
Disk:[Directory] : your:[directory]

ACKNOWLEDGEMENTS

We would like to thank Brian Kuhn for his initial effort in documenting the MRP system. We are also grateful to Louis Lapi, not only for his part in the original planning of the system but also for his participation in this document through helpful proposals on its format and contents.

APPENDIX I

Read Release Shell

```
      PROGRAM SEQREAD
      IMPLICIT INTEGER*4 (A-Z)
C
C     ILLUSTRATES HOW TO READ THE CWT RELEASE MASTER
C     IN A SEQUENTIAL MANNER IN TAGCODE SEQUENCE
C
C     $LINK,SEQREAD,CWT:[CWTSYS.TABLES]CWTLIB/LIB
C
C     PROGRAM GETS RELEASE RECORD PASSED TO HIM IN
C     COMMON BLOCK BELOW.
      INCLUDE 'CWT:[CWTSYS.RELEASE]RELRECCOM.FOR/LIST'
C
C     FIRST THIS CALL IS DONE TO OPEN THE FILE AND FIND
C     OUT HOW  MANY RELEASES THERE ARE...
C
      CALL CWTOPMST(NREC)
C
C     RECORD 1 IS A CONTROL RECORD.RELEASES LIVE IN
C     RECORDS 2 THRU NREC
C
      DO 10 I=2,NREC
C
C     GET A RELEASE....
C
      CALL CWTRELREADER(I)
      IF(DELSW.EQ.'Y') GO TO 10  !RELEASE FLAGGED
C                                ! INVALID
C
C     ********************************
C     *                              *
C     * INSERT YOUR CODE CODE HERE   *
C     * (EG,SELECTION CRITERIA,      *
C     *  ACCUMULATION RTNS)          *
C     *                              *
C     ********************************
C
10     CONTINUE
C
C     ********************************
C     *                              *
C     *   END OF FILE PROCESSING     *
C     *        GOES HERE             *
C     *                              *
C     ********************************
C
      END
```

APPENDIX II

Read Recovery Shell

```
      PROGRAM SHELL
      IMPLICIT INTEGER*4 (A-Z)
C
C     "SEQUENTIAL PASS" OF RECOVERY FILE FOR SOME YR...
C
C     DATA FOR INDIVIDUAL RECOVERIES PASSED VIA COMMON
C        BLOCK BELOW.
C
      INCLUDE 'CWT:[CWTSYS.RECOVERED]RECTAGCOM.FOR/LIST'

C
      DO 200 YR=Y1,Y2   !MAKE Y1,Y2 THE YEARS YOU WANT
C
C
      DO 100 I=1,999999
         OPT=1
         CALL CWTRECRD(YR,I,OPT,ERR)
         IF(ERR.NE.0) GO TO 150  !END THIS YEARS DATA
C
C*************************************************************
C        PUT YOUR SELECTION CODING,CALCULATIONS ETC IN
C              HERE
C
C*************************************************************
C
100   CONTINUE
C
150   CONTINUE  !END CURRENT YEAR
C
C*************************************************
C     PUT YOUR YEAR END CODE HERE
C*************************************************
C
C
200   CONTINUE  !END YEAR LOOP
C*************************************************************
C     PUT YOUR EOJ CODE HERE

C*************************************************************
900   CONTINUE
C
      STOP
      END
```

APPENDIX III

Combined Release and Catch Samples Shell

```
C_____________________________________________________________

            PROGRAM SHELL3
            IMPLICIT INTEGER*4 (A-Z)
      C
      C     "SEQUENTIAL PASS" OF RECOVERY FILE FOR SOME YR...
      C     ACCESS RELEASE AND CATCH/SAMPLE DATA ON SELECTED
      C     RECOVERIES.
      C     THIS CODE ILLUSTRATES THE USE OF SOME        C
     SUBROUTINES.IT IS
      C     NOT  GIVEN AS AN EXAMPLE OF THE MOST EFFICIENT
      C          ROUTE TO THE VARIOUS DATA SETS.
      C
      C     COMMON BLOCK FOR RECOVERY RECORDS:
            INCLUDE 'CWT:[CWTSYS.RECOVERED]RECTAGCOM.FOR/LIST'

      C
      C     COMMON BLOCK FOR RELEASE RECORDS:
            INCLUDE 'CWT:[CWTSYS.RELEASE]RELRECCOM.FOR/LIST'
      C
      C     DECLARATION FOR CATCH/SAMPLE DATA:
      C     (THE C/S DATA IS A PASSED ARGUMENT,NOT DONE VIA
      C      COMMON)
            INCLUDE 'CWT:[CWTSYS.CATSAM]CSRECCOM.FOR/LIST'
      C
            DO 200 YR=Y1,Y2   !MAKE Y1,Y2 THE YEARS U WANT
      C
      C
            DO 100 I=1,999999  !"LOOP DRIVER"  LOOP WILL EXIT
      C                                !   EARLIER
               OPT=1
               CALL CWTRECRD(YR,I,OPT,ERR)
               IF(ERR.NE.0) GO TO 150  !END THIS YEARS DATA
      C

C*************************************************************
      C        PUT YOUR SELECTION CODING,CALCULATIONS ETC IN
      C        HERE
      C        PRETEND WE WANT EXPANDABLE MARINE RECOVERIES OF
      C        CWT RELEASES.
      C
            IF(RECFORMAT.NE.1) GO TO 100  !WANT ONLY MARINE
      C                                   !RECOVERIES
      C
      C     LOOK SEE IF IT'S A "NON-TAG".(EG NO-PIN, LOST-PIN)
      C
            NOPT=3
            CALL CWTNONTAG(RECTAG,DUMMY,NOPT,NERR) !SEE
```

```
C                                                    !CWTNONTAG DOC
      IF(NERR.EQ.0) THEN    !IT IS A NON-TAG
         GO TO 100
      END IF
C
C
C     GET CATCH/SAMPLE DATA.
C
      CSOPT=1
      I4REGION=RECREGION  !SOME VARIABLES HAVE DIFFERENT
C                         !TYPES
C                         !SO - RECREGION IS I*2,RTN
C                         !WANTS I*4
      CALL CWTCSGET(I4REGION,YR,CSREC,CSOPT,CSERR)
      IF(CSERR.NE.0) THEN
        GO TO 100         !NO CATCH/SAMPLE AVAILABLE
      END IF
C
C
C     GET RELEASE DATA.
C
      ROPT=1   !HIT FILE BY TAGCODE
      CALL CWTRELGET(RECTAG,ROPT,RERR) !LOAD RELEASE
C                                      !DATA INTO COMMON
C
C
C
C   AT THIS POINT,PROGRAM HAS AVAILABLE A MARINE
C   RECOVERY,THE RELEASE
C   RECORD FOR THIS TAGCODE,AND THE CATCH/SAMPLE DATA
C   FOR THE CATCH
C   REGION AND YEAR THE RECOVERY OCCURRED IN.
C
C*************************************************************
C
100   CONTINUE
C
150   CONTINUE  !END CURRENT YEAR
C
C*************************************************
C     PUT YOUR YEAR END CODE HERE
C*************************************************
C
200   CONTINUE  !END YEAR LOOP
C
C*****************************************************************
C     PUT YOUR EOJ CODE HERE

C*****************************************************************
900   CONTINUE
C
      STOP
      END
```